

Final Report On:

**A Pressure-Based Finite Volume Method with Full Multi-Grid
Acceleration for Multi-Fluid Flow at All Speeds**

(F61775-00-WE071, SPC 00-4071)

Submitted to:

European Office of Aerospace Research and Development

(EOARD)

by

F. Moukalled

American University of Beirut,

Faculty of Engineering & Architecture,

Mechanical Engineering Department,
P.O.Box 11-0236
Beirut - Lebanon

Address all correspondence to F. Moukalled,

email: memouk@aub.edu.lb

DISTRIBUTION STATEMENT A

Approved for Public Release
Distribution Unlimited

September 30, 2001

20011025 082

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188		
<p>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>						
1. REPORT DATE (DD-MM-YYYY) 2001		2. REPORT TYPE Final Report		3. DATES COVERED (From - To) 9/12/00 - 12-Sep-01		
4. TITLE AND SUBTITLE A Pressure-Based Finite Volume Code with Full Multi-Grid Acceleration for Multi-Fluid Flow at All Speeds			5a. CONTRACT NUMBER F61775-00-WE071			
			5b. GRANT NUMBER			
			5c. PROGRAM ELEMENT NUMBER			
6. AUTHOR(S) Professor Fadi Moukalled			5d. PROJECT NUMBER			
			5d. TASK NUMBER			
			5e. WORK UNIT NUMBER			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) American University of Beirut ME Dept. PO Box: 11-0236 Beirut Lebanon			8. PERFORMING ORGANIZATION REPORT NUMBER N/A			
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) EOARD PSC 802 BOX 14 FPO 09499-0200			10. SPONSOR/MONITOR'S ACRONYM(S)			
			11. SPONSOR/MONITOR'S REPORT NUMBER(S) SPC 00-4071			
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.						
13. SUPPLEMENTARY NOTES						
14. ABSTRACT This report results from a contract tasking American University of Beirut as follows: Solving a variety of two-dimensional unsteady turbulent multi-fluid flow problems at all speeds (subsonic, transonic, and supersonic regimes) will test the newly implemented algorithms. Examples include: 1) Dam-type problem (fluid-air); 2) Gas-particle flow in a venturi meter; 3) Two fluid mixture at subsonic flow conditions (~incompressible); 4) Two gas mixture at supersonic flow conditions; 5) Multi-component turbulent combusting flows (i.e. chemical reaction with significant heat release).						
15. SUBJECT TERMS EOARD, Computational Fluid Dynamics (CFD)						
16. SECURITY CLASSIFICATION OF: a. REPORT UNCLAS		b. ABSTRACT UNCLAS	c. THIS PAGE UNCLAS	17. LIMITATION OF ABSTRACT UL	18. NUMBER OF PAGES 10088	19a. NAME OF RESPONSIBLE PERSON Charbel N. Raffoul
						19b. TELEPHONE NUMBER (Include area code) +44 (0)20 7514 4299

Abstract

This work is concerned with the implementation and testing of several segregated multi-fluid all speed flow algorithms. These algorithms are part of the Mass Conservation Based Algorithms (MCBA) group in which the pressure correction equation is derived from overall mass conservation. Seven MCBA algorithms (based on SIMPLE, SIMPLEC, SIMPLEX, SIMPLEM, SIMPLEST, PISO, and PRIME) are investigated within a structured collocated non-linear multi-grid finite-volume framework. The performance and accuracy of these algorithms are assessed by solving a variety of one- and two-dimensional laminar and turbulent two-phase flow problems in the subsonic, transonic, and supersonic regimes and comparing results with published numerical and/or experimental data. The SG method is used to solve for the one-dimensional test problems and the effects of grid size on convergence characteristics are analyzed. However, solutions for the two-dimensional problems are generated for several grid systems using the single grid method (SG), the prolongation grid method (PG), and the full non-linear multi-grid method (FMG) and their effects on convergence behavior are studied. The main outcomes of this study are the clear demonstrations of: (i) the capability of all MCBA algorithms to deal with multi-fluid flow situations when the fluidic continuity equations are normalized; (ii) the ability of the FMG method to tackle the added non-linearity of multi-fluid flows, which is manifested through the performance jump observed when using the non-linear multi-grid as compared to the SG and PG methods; (iii) the extension of the FMG method to predict turbulent multi-fluid flows; (iv) the capacity of the MCBA algorithms to predict multi-fluid flow at all speeds. Moreover, results displayed in terms of convergence history plots and CPU-times, indicate that the performances of SIMPLE, SIMPLEC, and SIMPLEX are very close. In general, the performance of SIMPLEST approaches that of SIMPLE for diffusion-dominated flows (i.e. turbulent flows). As expected, the PRIME algorithm is found to be the most expensive due to the explicit treatment of the fluidic momentum equations. The PISO algorithm is generally more expensive than SIMPLE and its performance depends on the type of flow and solution method used. The behavior of SIMPLEM is consistent and in terms of CPU effort it stands between PRIME and SIMPLE. For all algorithms, the use of the PG and FMG methods accelerate the convergence rate. The FMG method is by far more

efficient than the PG method, accelerating the convergence rate, for the problems solved and the grid used, over the SG method by a factor reaching a value as high as 10 and undoubtedly verifying the effectiveness of multi-grid methods in tackling the linearity of multi-phase flows.

Nomenclature

$A_p^{(k)}$, ..	coefficients in the discretized equation for $\phi^{(k)}$.
$B_p^{(k)}$	source term in the discretized equation for $\phi^{(k)}$.
$\mathbf{B}^{(k)}$	body force per unit volume of fluid/phase k.
$C_p^{(k)}$	coefficient equals to $1/R^{(k)}T^{(k)}$.
$\hat{\mathbf{d}}_f$	covariant unit vector (i.e. in the direction of \mathbf{d}_f).
$D_p^{(k)}[\phi^{(k)}]$	the D operator.
$\mathbf{D}_p^{(k)}[\phi^{(k)}]$	the vector form of the D operator.
$g^{(km)}$	drag function (Eq. (4)).
$H_p[\phi^{(k)}]$	the H operator.
$HI_p[\phi^{(k)}]$	the HI operator working on $\phi^{(k)}$ ($\phi^{(k)}=u^{(k)}, v^{(k)}$, or $w^{(k)}$).
$HP_p[\phi^{(k)}]$	the HP operator working on $\phi^{(k)}$ ($\phi^{(k)}=u^{(k)}, v^{(k)}$, or $w^{(k)}$).
$\mathbf{H}\mathbf{P}_p[u^{(k)}]$	the vector form of the HP operator.
$\mathbf{H}\mathbf{I}_p[u^{(k)}]$	the vector form of the HI operator.
$\mathbf{I}^{(k)}$	inter-phase momentum transfer.
$\mathbf{J}_f^{(k)D}$	diffusion flux of $\phi^{(k)}$ across cell face 'f'.
$\mathbf{J}_f^{(k)C}$	convection flux of $\phi^{(k)}$ across cell face 'f'.
$\dot{M}^{(k)}$	mass source per unit volume.
$\tilde{\mathbf{n}}_f$	contravariant unit vector (i.e. in the direction of \mathbf{S}_f).
P	pressure.
$Pr^{(k)}, Pr_t^{(k)}$	laminar and turbulent Prandtl number for fluid/phase k.
$\dot{q}^{(k)}$	heat generated per unit volume of fluid/phase k.
$Q^{(k)}$	general source term of fluid/phase k.
$r^{(k)}$	volume fraction of fluid/phase k.
$R^{(k)}$	gas constant for fluid/phase k.
\mathbf{S}_f	surface vector.
t	time.
$T^{(k)}$	temperature of fluid/phase k.
$U_f^{(k)}$	interface flux velocity $(\mathbf{v}_f^{(k)} \cdot \mathbf{S}_f)$ of fluid/phase k.

$\mathbf{u}^{(k)}$	velocity vector of fluid/phase k.
$u^{(k)}, v^{(k)}, \dots$	velocity components of fluid/phase k.
x, y, z	Cartesian coordinates.
$\ a, b\ $	the maximum of a and b.

Greek Symbols

$\rho^{(k)}$	density of fluid/phase k.
$\Gamma^{(k)}$	diffusion coefficient of fluid/phase k.
$\Phi^{(k)}$	dissipation term in energy equation of fluid/phase k.
$\phi^{(k)}$	general scalar quantity associated with fluid/phase k.
κ_f	space vector equal to $(\hat{\mathbf{n}}_f - \gamma \hat{\mathbf{d}}_f) S_f$
$\Delta_P [\phi^{(k)}]$	the Δ operator.
$\mu^{(k)}, \mu_t^{(k)}$	laminar and turbulent viscosity of fluid/phase k.
Ω	cell volume.
$\beta^{(k)}$	thermal expansion coefficient for phase/fluid k.
δt	time step.
γ	scaling factor.

Subscripts

e, w, .	refers to the east, west, ... face of a control volume.
E, W, ..	refers to the East, West, ... neighbors of the main grid point.
f	refers to control volume face f.
P	refers to the P grid point.

Superscripts

C	refers to convection contribution.
D	refers to diffusion contribution.
(k)	refers to fluid/phase k.
$(k)^*, (k)^{**}, \dots$	refers to first, second, ... updated value at the current iteration.
$(k)^\circ$	refers to values of fluid/phase k from the previous iteration.
$(k)'$	refers to correction field of phase/fluid k.
m	refers to fluid/phase m.
old	refers to values from the previous time step.
sx	refers to SIMPLEX.

Introduction

The extensive developments that have taken place in Computational Fluid Dynamics (CFD) during the last three decades have established this evolving technology as a reliable and critical tool for the simulation of a wide variety of engineering fluid flow processes (mixing, solidification, turbulence, ...). During the past few years several issues have been addressed. Concerns related to accuracy were assuaged through the development of High Resolution (HR) schemes [1-13]. Moreover, better solution algorithms [14-20], solvers [21,22], and multi-grid techniques [23-26] have greatly reduced the computational cost and made it feasible to solve real life problems. Furthermore, pressure-based algorithms [27-40] were extended to solve flow problems in the various Reynolds and Mach number regimes using both structured and unstructured grid methods [41-46].

While high-resolution schemes, solvers, multi-grid techniques, structured and unstructured grid methods, etc... can be applied indiscriminately to the simulation of single-fluid or multi-fluid flow, nearly all developments in solution algorithms have been directed towards the simulation of single-fluid flow [24,26,47,48]. In particular, many of single-fluid segregated solution algorithms were developed such as the well known SIMPLE [14,15], the SIMPLEST [49], the SIMPLEC [17], the SIMPLEM [50], the PISO [16], the PRIME [51], and the SIMPLEX [19] algorithms, to site a few. Additionally, several techniques have been devised to improve the performance, facilitate the implementation, and extend the capability of these algorithms.

Developments in solution algorithms for simulating multi-fluid phenomena have, on the other hand, lagged behind that of single-fluid algorithms due to the much higher computational cost involved, to numerical difficulties that had to be first addressed in the simulation of single-fluid flow, and to the increase in algorithmic complexity. While the major difficulty in the simulation of single-fluid flow stems from the coupling between the momentum and continuity equations, in the simulation of multi-fluid flow phenomena, this problem is further complicated by the fact that there are as many sets of continuity and momentum equations as there are fluids, that they are all coupled together in various ways (interchange momentum by inter-phase mass and momentum transfer, etc.) and that the fluids share space (the volume fractions sum to unity, but are not known in advance).

Despite these complexities, successful segregated pressure-based solution algorithms have been devised. The IPSA variants devised by the Spalding Group at Imperial College [52-54] and the set of algorithms devised by the Los Alamos Scientific Laboratory (LASL) group [55-57] are examples of multiphase algorithms. However, in contrast with the widespread information available on single-fluid solution algorithms, little information is available about multi-fluid solution algorithms, a fact that has confined their use to a small community, slowed their development, and isolated them from newer developments in single-fluid flow algorithms (PWIM, all speed flows,...).

Recently, Darwish et al. [58,59] extended the segregated single-fluid flow algorithms [20] to predict multi-fluid flow at all speeds and derived these algorithms using a unified, compact, and easy to understand notation. Furthermore, it was shown in [58,59] that the pressure correction equation can be derived either by using the geometric conservation equation or the overall mass conservation equation. Depending on which equation is used, the segregated pressure-based multi-fluid flow algorithms were classified respectively as either the Geometric Conservation Based family of Algorithms (MCBA) or the Mass Conservation Based family of Algorithms (GCBA).

The objective of the present work is to test and compare the different algorithms in the MCBA group and to assess their performance by solving various one- and two-dimensional laminar and turbulent two-phase problems in the subsonic, transonic, and supersonic regimes. In the implementation of these algorithms care is taken to use a high accuracy discretization within the framework of a non-linear full multi-grid solution procedure [60-66].

In what follows, the governing equations are first introduced, followed by a brief description of the discretization procedure. Then the MCBA algorithms are presented, their capabilities to predict multi-fluid flow phenomena at all speeds demonstrated, and their performance characteristics (in terms of convergence history and speed) assessed. For that purpose, a total of twelve laminar and turbulent incompressible and compressible problems encompassing dilute and dense gas-solid, and bubbly flows in the subsonic, transonic and supersonic regimes are solved. In addition, the performance of these algorithms on a single grid and within a multi-grid environment is compared.

The Governing Equations

In multi-fluid flow the various fluids/phases coexist with different concentrations at different locations in the flow domain and move with unequal velocities. Thus, the equations governing multi-fluid flows are the conservation laws of mass, momentum, and energy for each individual fluid in addition to a set of auxiliary relations.

Conservation of mass

The volume fraction $r^{(k)}$, which is the portion of volumetric space occupied by the k^{th} fluid ($\Omega^{(k)}/\Omega$) along with the k^{th} fluid density, $\rho^{(k)}$, and velocity, $\mathbf{u}^{(k)}$, in order to satisfy the mass-conservation principle, have to obey the differential equation:

$$\frac{\partial(r^{(k)}\rho^{(k)})}{\partial t} + \nabla \cdot (r^{(k)}\rho^{(k)}\mathbf{u}^{(k)}) = r^{(k)}\dot{M}^{(k)} \quad (1)$$

Mass sources are often non-zero, as when one fluid is transformed to another fluid. However, summation over all fluids leads to the following “global mass-conservation” equation:

$$\sum_k \left(\frac{\partial(r^{(k)}\rho^{(k)})}{\partial t} + \nabla \cdot (r^{(k)}\rho^{(k)}\mathbf{u}^{(k)}) \right) = 0 \quad (2)$$

The zero on the right-hand side signifies that the sum of mass sources (generation and loss) is zero.

Conservation of momentum

Denoting the velocity of the k^{th} phase by $\mathbf{u}^{(k)}$, then the fluidic momentum equation (for the k^{th} phase) can be written as:

$$\frac{\partial(r^{(k)}\rho^{(k)}\mathbf{u}^{(k)})}{\partial t} + \nabla \cdot (r^{(k)}\rho^{(k)}\mathbf{u}^{(k)}\mathbf{u}^{(k)}) = \nabla \cdot [r^{(k)}(\mu^{(k)} + \mu_t^{(k)})\nabla \mathbf{u}^{(k)}] + r^{(k)}(-\nabla P + \mathbf{B}^{(k)}) + \mathbf{I}_M^{(k)} \quad (3)$$

Here P stands for the pressure, which is assumed to be shared by all fluids, $\mathbf{B}^{(k)}$ is the body force per unit volume of phase (k) , $\mu^{(k)}$ and $\mu_t^{(k)}$ are the laminar and turbulent viscosity, and $\mathbf{I}_M^{(k)}$ represents the interfacial forces per unit volume due to drag, virtual mass effects, lift, etc. Because of the many suggested closures for the various interfacial sources, the ones adopted in the present study will be presented as needed.

Conservation of energy

Let $T^{(k)}$ be the temperature of the k^{th} phase, the fluidic energy equation for the k^{th} fluid is then given by:

$$\frac{\partial(r^{(k)}\rho^{(k)}T^{(k)})}{\partial t} + \nabla \cdot (r^{(k)}\rho^{(k)}\mathbf{u}^{(k)}T^{(k)}) = \nabla \cdot \left[r^{(k)} \left(\frac{\mu^{(k)}}{Pr^{(k)}} + \frac{\mu_t^{(k)}}{Pr_t^{(k)}} \right) \nabla T^{(k)} \right] + \frac{r^{(k)}}{c_p^{(k)}} \left\{ \beta^{(k)} T^{(k)} \left[\frac{\partial P}{\partial t} + \nabla \cdot (P\mathbf{u}^{(k)}) - P\nabla \cdot (\mathbf{u}^{(k)}) \right] + \Phi^{(k)} + \dot{Q}^{(k)} \right\} + \frac{I_E^{(k)}}{c_p^{(k)}} \quad (4)$$

where $\Phi^{(k)}$ is the viscous dissipation function, $\beta^{(k)}$ the thermal expansion coefficient which is equal to $1/T^{(k)}$ for an ideal gas, $Pr^{(k)}$ and $Pr_t^{(k)}$ are the laminar and turbulent Prandtl numbers, and $I_E^{(k)}$ is the interfacial energy transfer to phase (k).

Turbulence Modeling

The effect of turbulence on interfacial mass, momentum, and energy transfer is difficult to model and is still an active area of research. Similar to single-fluid flow, researchers have advertised several flow-dependent models to describe turbulence. These models vary in complexity from simple algebraic [67] models to state-of-the-art Reynolds-stress [68] models. However, the widely used multi-phase turbulence model, presented next, is the two-equation $k-\epsilon$ model [69]. Without going into details, the fluidic conservation equations governing the turbulence kinetic energy (k) and turbulence dissipation rate (ϵ) for the k^{th} fluid are given by:

$$\frac{\partial(r^{(k)}\rho^{(k)}k^{(k)})}{\partial t} + \nabla \cdot (r^{(k)}\rho^{(k)}\mathbf{u}^{(k)}k^{(k)}) = \nabla \cdot \left(r^{(k)} \frac{\mu_t^{(k)}}{\sigma_k^{(k)}} \nabla k^{(k)} \right) + r^{(k)}\rho^{(k)}(G^{(k)} - \epsilon^{(k)}) + I_k^{(k)} \quad (5)$$

$$\frac{\partial(r^{(k)}\rho^{(k)}\epsilon^{(k)})}{\partial t} + \nabla \cdot (r^{(k)}\rho^{(k)}\mathbf{u}^{(k)}\epsilon^{(k)}) = \nabla \cdot \left(r^{(k)} \frac{\mu_t^{(k)}}{\sigma_\epsilon^{(k)}} \nabla \epsilon^{(k)} \right) + r^{(k)}\rho^{(k)} \frac{\epsilon^{(k)}}{k^{(k)}} (c_{1\epsilon} G^{(k)} - c_{2\epsilon} \epsilon^{(k)}) + I_\epsilon^{(k)} \quad (6)$$

where $I_k^{(k)}$ and $I_\epsilon^{(k)}$ represent the interfacial turbulence terms. The turbulent viscosity is calculated as:

$$\mu_t^{(k)} = C_\mu \frac{[k^{(k)}]^2}{\epsilon^{(k)}} \quad (7)$$

For two-phase flows, several extensions of the k- ϵ model that are based on calculating the turbulent viscosity by solving the k and ϵ equations for the carrier or continuous phase only have been proposed in the literature [70-76]. In a recent article, Cokljat and Ivanov [69] presented a phase coupled k- ϵ turbulence model, intended for the cases where a non-dilute secondary phase is present, in which the k- ϵ transport equations for all phases are solved. Since the method is still not well developed, the first approach in which only the k and ϵ equations for the carrier phase are solved is adopted in this work. Details regarding the specific model used will be presented as needed.

The General Fluidic Scalar Equation

A review of the above differential equations reveals that they are similar in structure. If a typical representative variable associated with phase (k) is denoted by $\phi^{(k)}$, the general fluidic differential equation may be written as:

$$\frac{\partial(r^{(k)}\rho^{(k)}\phi^{(k)})}{\partial t} + \nabla \cdot (r^{(k)}\rho^{(k)}\mathbf{u}^{(k)}\phi^{(k)}) = \nabla \cdot (r^{(k)}\Gamma^{(k)}\nabla\phi^{(k)}) + r^{(k)}Q^{(k)} \quad (8)$$

where the expression for $\Gamma^{(k)}$ and $Q^{(k)}$ can be deduced from the parent equations.

Auxiliary Relations

The above set of differential equations has to be solved in conjunction with constraints on certain variables represented by algebraic relations. These auxiliary relations include the equations of state, the geometric conservation equation, and the interfacial mass, momentum, energy, and turbulence energy transfers.

Physically, the geometric conservation equation is a statement indicating that the sum of volumes occupied by the different fluids within a cell is equal to the volume of the cell containing the fluids.

$$\sum_k r^{(k)} = 1 \quad (9)$$

For a compressible multi-fluid flow, auxiliary equations of state relating density to pressure and temperature are needed. For the k^{th} phase, such an equation can be written as:

$$\rho^{(k)} = \rho^{(k)}(P, T^{(k)}) \quad (10)$$

Several models have been developed for computing the interfacial mass, momentum, energy, and turbulence energy transfers terms. The closures used in this work will be detailed whenever they arise while solving problems.

In order to present a complete mathematical problem, thermodynamic relations may be needed and initial and boundary conditions should supplement the above equations.

Discretization Procedure

In the previous sections the differential equations governing multi-fluid flow phenomena were outlined as well as the associated auxiliary relations. The task now is to present the Finite Volume-based numerical solution algorithm for solving these equations.

Discretization of the general fluidic conservation equation

The general conservation equation (8) is integrated over a finite volume to yield:

$$\begin{aligned} \iint_{\Omega} \frac{\partial(r^{(k)} \rho^{(k)} \phi^{(k)})}{\partial t} d\Omega + \iint_{\Omega} \nabla \cdot (r^{(k)} \rho^{(k)} \mathbf{u}^{(k)} \phi^{(k)}) d\Omega \\ = \iint_{\Omega} \nabla \cdot (r^{(k)} \Gamma^{(k)} \nabla \phi^{(k)}) d\Omega + \iint_{\Omega} r^{(k)} Q^{(k)} d\Omega \end{aligned} \quad (11)$$

Where Ω is the volume of the control cell. Using the divergence theorem to transform the volume integral into a surface integral and then replacing the surface integral by a summation of the fluxes over the sides of the control volume, equation (11) is transformed to:

$$\frac{\partial(r^{(k)} \rho^{(k)} \phi^{(k)} \Omega)}{\partial t} + \sum_{nb=e,w,n,s,t,b} (\mathbf{J}_{nb}^{(k)D} + \mathbf{J}_{nb}^{(k)C}) = r^{(k)} Q^{(k)} \Omega \quad (12)$$

where $\mathbf{J}_{nb}^{(k)D}$ and $\mathbf{J}_{nb}^{(k)C}$ are the diffusive and convective fluxes, respectively. The discretized form of the diffusive flux along an east face is given by:

$$\mathbf{J}_e^{(k)D} = -r_e^{(k)} \Gamma_e^{(k)} \left[(\phi_E^{(k)} - \bar{\phi}_P^{(k)}) \frac{\mathbf{S}_e \cdot \mathbf{S}_e}{\mathbf{S}_e \cdot \mathbf{d}_e} + \overline{(\nabla \phi^{(k)})}_e \cdot \boldsymbol{\kappa}_e \right] \quad (13)$$

where the over bar denotes a value obtained by interpolation, $\boldsymbol{\kappa}_e$ is a space vector defined as,

$$\boldsymbol{\kappa}_e = \left(\hat{\mathbf{n}}_e - (\gamma \hat{\mathbf{d}})_e \right) \mathbf{S}_e = \kappa_e^x \mathbf{i} + \kappa_e^y \mathbf{j} + \kappa_e^z \mathbf{k} \quad (14)$$

and γ is a scaling factor given by [77],

$$\gamma_e = \frac{1}{\hat{\mathbf{n}}_e \cdot \hat{\mathbf{d}}_e} = \frac{\mathbf{S}_e \cdot \mathbf{d}_e}{\mathbf{S}_e \cdot \hat{\mathbf{d}}_e} \quad (15)$$

and $\hat{\mathbf{n}}_e$ and $\hat{\mathbf{d}}_e$ are the contravariant and covariant unit vectors respectively. The discretized convective flux along an east face is given by:

$$J_e^{(k)C} = (\mathbf{r}^{(k)} \rho^{(k)} \mathbf{u}^{(k)} \cdot \mathbf{S})_e \phi_e^{(k)} = (\mathbf{r}^{(k)} \rho^{(k)} \mathbf{U}^{(k)})_e \phi_e^{(k)} = C_e^{(k)} \phi_e^{(k)} \quad (16)$$

where \mathbf{S}_e and C_e are the surface vector and convection flux coefficient at cell face 'e', respectively. As can be seen from Eq. (16), the accuracy of the control volume solution for the convective scalar flux depends on the proper estimation of the face value ϕ_e as a function of the neighboring ϕ nodes values. Using some assumed interpolation profile, ϕ_e can be explicitly formulated in terms of the nodal values by a functional relationship of the form:

$$\phi_e = f(\phi_{nb}) \quad (17)$$

where ϕ_{nb} denotes the ϕ values at the neighboring nodes. The interpolation profile may be one-dimensional or multi-dimensional of low or high order of accuracy. The higher the order of the profile is, the lower numerical diffusion will be. However, the order of the profile and its dimensionality do not eliminate numerical dispersion. To minimize this error, limiters on the convective flux should be imposed to enforce monotonicity. The flux limiter denoted by the Convective Boundedness Criterion (CBC) [2] is adopted here and applied within the context of the Normalized Variable and Space Formulation methodology (NVSF) [8] on a quadratic upwind biased profile, which is third order accurate, to produce the SMART [2] convective scheme. Unless otherwise stated, the High Resolution (HR) SMART scheme is used in solving all problems presented here. For more details regarding the NVSF methodology the reader is referred to [8].

After substituting the face values by their functional relationship relating to the node values of ϕ , Eq. (12) is transformed after some algebraic manipulations into the following discretized equation:

$$A_p^{(k)} \phi_p^{(k)} = \sum_{NB} A_{NB}^{(k)} \phi_{NB}^{(k)} + B_p^{(k)} \quad (18)$$

where the coefficients $A_p^{(k)}$ and $A_{NB}^{(k)}$ depend on the selected scheme and $B_p^{(k)}$ is the source term of the discretized equation. In compact form, the above equation can be written as

$$\phi^{(k)} = H_p[\phi^{(k)}] = \frac{\sum_{NB} A_{NB}^{(k)} \phi_{NB}^{(k)} + B_p^{(k)}}{A_p^{(k)}} \quad (19)$$

Discretization of the fluidic momentum equation

The discretization procedure for the momentum equation yields an algebraic equation of the form:

$$A_p^{(k)} u_p^{(k)} = \sum_{NB(P)} A_{NB}^{(k)} u_{NB}^{(k)} + B_p^{(k)} - r_p^{(k)} \Omega_p \nabla_p(P) + \Omega_p \sum_{m \neq k} g^{(km)} (u_p^{(m)} - u_p^{(k)}) \quad (20)$$

In the above equation, the inter-phase term is written out explicitly to show the strong coupling among the momentum equations of the different fluids. This is different from the *spatial* coupling that exists among the neighboring velocities of the same fluid. To improve the overall convergence and robustness of the algorithm, the inter-phase source term can be linearized and the linear part treated implicitly so that:

$$A_p^{(k)} u_p^{(k)} = \sum_{NB} A_{NB}^{(k)} u_{NB}^{(k)} + B_p^{(k)} - r_p^{(k)} \Omega_p \nabla_p(P) + \Omega_p \sum_{m \neq k} g^{(km)} u_p^{(m)} \quad (21)$$

Where now

$$A_p^{(k)} \leftarrow A_p^{(k)} + \Omega_p \sum_{m \neq k} (g_p^{(km)}) \quad (22)$$

For later reference, the discretized form of the momentum equation is written in the following two forms:

$$u_p^{(k)} = H P_p[u^{(k)}] - r^{(k)} D_p^{(k)} \nabla_p(P) \quad (23)$$

where the body force and inter-phase terms are absorbed in the $B_p^{(k)}$ source term within the $H P_p[u^{(k)}]$ term, or as

$$u_p^{(k)} = H I_p[u^{(k)}] + D_p^{(k)} \sum_{m \neq k} (g^{(km)} u_p^{(m)}) \quad (24)$$

where the body force and pressure gradient terms are absorbed in the $B_p^{(k)}$ source term within the $H I_p[u^{(k)}]$ term.

Discretization of the fluidic mass conservation equation

The fluidic mass-conservation equation (Eq. (1)) can be viewed as a fluidic volume fraction equation:

$$r_p^{(k)} = H_p[r^{(k)}] \quad (25)$$

or as a fluidic continuity equation to be used in deriving the pressure correction equation:

$$\frac{(r_p^{(k)} \rho_p^{(k)}) - (r_p^{(k)} \rho_p^{(k)})^{\text{old}}}{\delta t} \Omega + \Delta_p [r^{(k)} \rho^{(k)} \mathbf{u}^{(k)} \cdot \mathbf{S}] = r^{(k)} \dot{M}^{(k)} \quad (26)$$

where the ? operator represents the following operation:

$$\Delta_p[\Theta] = \sum_{f=NB(P)} \Theta_f \quad (27)$$

Discretization of the fluidic energy equation

The discretization of the energy equation follows that of the general fluidic scalar equation. The only difference is the one pertaining to the discretization of the additional source terms. Since a control volume approach is followed, the integral of these sources over the control volume appears in the discretized equation. By using the divergence theorem, the volume integral is transformed into a surface integral and the resultant discretized expressions evaluated explicitly.

The Non-linear Multi-Grid Strategy

Standard relaxation techniques reduce those Fourier components of the error whose wavelengths are comparable with the mesh size, and then performance degrades when dealing with the low-frequency components of the error. The underlying idea of the multi-grid strategy is to use progressively coarser grids on which the low-frequency error components on the finest grid appear as high-frequency Fourier mode for which the relaxation scheme works efficiently. In the present work, this strategy is adopted to accelerate convergence and thereby reduce the overall computational cost.

The full multi-grid cycle adopted in this work can be summarized as follows. The coarser grids are constructed by agglomerating four fine grid cells in a square topology. The algorithm starts at the coarsest level and the solved fields are interpolated onto the finer mesh and used as initial guess. Few iterations are performed on the fine mesh before the solution is restricted back to the coarser mesh to be used as an initial guess for the solution at this level. A forcing term is added to the discrete conservation equations on the coarser grid to ensure that the equation at the coarse grid represents the fine grid equation. After performing a number of iterations on the coarse mesh, the solution is prolonged onto the finer mesh and a number of iterations are completed. This process continues until convergence. The solution is then interpolated into a finer mesh and the process repeated until

convergence is reached on the desired finest mesh. This strategy has been applied to both incompressible and compressible multi-fluid flows.

The Mass Conservation Based Algorithms

The number of equations describing an n -fluid flow situation are: n fluidic momentum equations, n fluidic volume fraction (or mass conservation) equations, a geometric conservation equation, and for the case of a compressible flow an additional n auxiliary pressure-density relations. Moreover, the variables involved are the n fluidic velocity vectors, the n fluidic volume fractions, the pressure field, and for a compressible flow an additional n unknown fluidic density fields. In all MCBA algorithms, the n momentum equations are used to calculate the n velocity fields, $n-1$ volume fraction (mass conservation) equations are used to calculate $n-1$ volume fraction fields, and the last volume fraction field calculated using the geometric conservation equation

$$r^{(n)} = 1 - \sum_{k \neq n} r^{(k)} \quad (28)$$

The remaining volume fraction equation can be used to calculate the pressure field. However, instead of using this last volume fraction equation, the global conservation equation can be employed, i.e. the sum of the individual mass conservation equations, to derive a pressure correction equation. The sequence of events in the MCBA is as follows:

1. Solve the fluidic momentum equations for velocities.
2. Solve the pressure correction equation based on global mass conservation.
3. Correct velocities, densities, and pressure.
4. Solve the fluidic mass conservation equations for volume fractions.
5. Solve fluidic the energy equations.
6. Return to the first step and repeat until convergence.

Solving for Velocities

In this first step, the fluidic momentum equations are solved to find $\mathbf{u}_p^{(k)*}$ based on guessed or previously calculated volume fraction and pressure fields:

$$\mathbf{u}_P^{(k)} = \mathbf{H}\mathbf{I}_P[\mathbf{u}^{(k)}] + \mathbf{D}_P^{(k)} \sum_{m \neq k} g^{(km)} \mathbf{u}_P^{(m)} \quad (29)$$

From Eq. (29) it is clear that the $\mathbf{H}\mathbf{I}_P[\mathbf{u}^{(k)}]$ term couples $\mathbf{u}_P^{(k)}$ to the neighbouring phase (k) velocities (*geometric or spatial coupling*) while the $\mathbf{D}_P^{(k)} \sum_{m \neq k} g^{(km)} \mathbf{u}_P^{(m)}$ term couples $\mathbf{u}_P^{(k)}$ to the velocity of all other phases at grid point P (*inter-phase coupling*). Therefore, the rate of convergence of the iterative solution procedure used to solve the above system will greatly depend on its capability to resolve both types of coupling. The spatial coupling presents no problem to the well-established iterative techniques since it couples velocities of the same phase. The inter-phase coupling is however problematic since it relates velocities of different phases. An explicit evaluation of this term slows the convergence rate considerably especially when the inter-fluid momentum transfer terms, represented by $g^{(mn)}$, are large. To accelerate convergence, the Partial Elimination Algorithm (PEA)[78] or the Simultaneous solution of Non-linearly Coupled Equations technique (SINCE) [79] is used.

Solving for Pressure Correction

To derive the pressure-correction equation, the mass conservation equations of the various phases are added to yield the global mass conservation equation given by:

$$\sum_k \left\{ \frac{(r_P^{(k)} p_P^{(k)}) - (r_P^{(k)} p_P^{(k)})^{\text{old}}}{\delta t} \Omega + \Delta_p (r^{(k)} p_P^{(k)} \mathbf{u}^{(k)} \cdot \mathbf{S}) \right\} = \sum_k r^{(k)} \dot{M}^{(k)} = 0 \quad (30)$$

In the predictor stage a guessed or an estimated pressure field from the previous iteration, denoted by P° , is substituted into the momentum equations. The resulting velocity fields denoted by $\mathbf{u}^{(k)*}$ which now satisfy the momentum equations will not, in general, satisfy the mass conservation equations. Thus, corrections are needed in order to yield velocity and pressure fields that satisfy both equations. Denoting the corrections for pressure, velocity, and density by P' , $\mathbf{u}^{(k)'}$, and $p^{(k)'}$ respectively, the corrected fields are written as:

$$P = P^\circ + P', \mathbf{u}^{(k)} = \mathbf{u}^{(k)*} + \mathbf{u}^{(k)'}, p^{(k)} = p^{(k)\circ} + p^{(k)'} \quad (31)$$

Hence the equations solved in the predictor stage are:

$$\mathbf{u}_P^{(k)*} = \mathbf{H}\mathbf{P}_P[\mathbf{u}^{(k)*}] - r^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_p P^\circ \quad (32)$$

While the final solutions satisfy

$$\mathbf{u}_p^{(k)} = \mathbf{H}\mathbf{P}_p[\mathbf{u}^{(k)}] - r^{(k)\circ} \mathbf{D}_p^{(k)} \nabla_p P \quad (33)$$

Subtracting the two equation sets ((33) and (32)) from each other yields the following equation involving the correction terms:

$$\mathbf{u}_p^{(k)'} = \mathbf{H}\mathbf{P}_p[\mathbf{u}^{(k)'}] - r^{(k)\circ} \mathbf{D}_p^{(k)} \nabla_p P' \quad (34)$$

Moreover, the new density and velocity fields, $\rho^{(k)}$ and $\mathbf{u}^{(k)}$, will satisfy the overall mass conservation equation if:

$$\sum_k \left\{ \frac{(r_p^{(k)\circ} \rho_p^{(k)}) - (r_p^{(k)} \rho_p^{(k)})^{\text{Old}}}{\delta t} \Omega + \Delta_p [r^{(k)\circ} \rho^{(k)} \mathbf{u}^{(k)} \cdot \mathbf{S}] \right\} = 0 \quad (35)$$

Linearizing the $(\rho^{(k)} \mathbf{u}^{(k)})$ term, one gets

$$(\rho^{(k)*} + \rho^{(k)'}) (\mathbf{u}^{(k)*} + \mathbf{u}^{(k)'}) = \rho^{(k)*} \mathbf{u}^{(k)*} + \rho^{(k)*} \mathbf{u}^{(k)'} + \rho^{(k)'} \mathbf{u}^{(k)*} + \rho^{(k)'} \mathbf{u}^{(k)'} \quad (36)$$

Substituting equations (36) and (34) into equation (35), rearranging, and replacing density correction by pressure correction, the final form of the pressure-correction equation is written as:

$$\begin{aligned} \sum_k \left\{ \frac{\Omega}{\delta t} r_p^{(k)\circ} C_p^{(k)} P'_p + \Delta_p [r^{(k)\circ} U^{(k)*} C_p^{(k)} P'] - \Delta_p [r^{(k)\circ} \rho^{(k)*} (r^{(k)\circ} \mathbf{D}^{(k)} \nabla P') \mathbf{S}] \right\} \\ = - \sum_k \left\{ \frac{r_p^{(k)\circ} \rho_p^{(k)*} - (r_p^{(k)} \rho_p^{(k)})^{\text{Old}}}{\delta t} \Omega + \Delta_p [r^{(k)\circ} \rho^{(k)*} U^{(k)*}] \right\} \\ + \Delta_p [r^{(k)\circ} \rho^{(k)*} (\mathbf{H}\mathbf{P}[\mathbf{u}^{(k)'}] \mathbf{S})] + \Delta_p [r^{(k)\circ} \rho^{(k)'} \mathbf{u}^{(k)'} \cdot \mathbf{S}] \end{aligned} \quad (37)$$

The second order correction term $\rho^{(k)'} \mathbf{u}^{(k)'}$ is usually neglected. Moreover, if the $\mathbf{H}\mathbf{P}[\mathbf{u}^{(k)'}]$ term in the above equation is retained, there will result a pressure correction equation relating the pressure correction value at a point to all values in the domain. To facilitate implementation and reduce cost, simplifying assumptions related to this term have been introduced. Depending on these assumptions, different algorithms are obtained. A summary of the various GCBA algorithms used in this work is given next.

The MCBA's in Symbolic Form

In what follow, the multi-fluid versions of the single fluid SIMPLE [14,15], SIMPLEC [17], SIMPLEST [49], SIMPLEX [19], SIMPLEM [50], PISO [16], and PRIME [51] segregated algorithms are given using a symbolic format. In the notation used, the superscripts “old” and “o” denote values

from the previous time step and values from the previous iteration, respectively. Moreover, the superscripts *, **, ***, and **** represent the first, second, third, and fourth updated values at the current iteration, respectively.

The MCBA following SIMPLE (MCBA-SIMPLE): Symbolic Form

Predictor:

$$\mathbf{u}_P^{(k)*} = \mathbf{HP}_P[\mathbf{u}^{(k)*}] - r^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_P P^* \quad (38)$$

Corrector:

$$(\mathbf{u}^{(k)'}, P', \rho^{(k)'}) (\mathbf{u}^{(k)**} = \mathbf{u}^{(k)*} + \mathbf{u}^{(k)'}, P^* = P^* + P', \rho^{(k)*} = \rho^{(k)\circ} + \rho^{(k)'}) \quad (39)$$

$$\therefore \mathbf{u}_P^{(k)**} = \mathbf{HP}_P[\mathbf{u}^{(k)**}] - r^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_P P^* = \mathbf{HP}_P[\mathbf{u}^{(k)*} + \mathbf{u}^{(k)'}] - r^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_P (P^* + P') \quad (40)$$

$$\therefore \begin{cases} \mathbf{u}_P^{(k)'} = \mathbf{HP}_P[\mathbf{u}^{(k)'}] - r^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_P P' \\ \rho^{(k)'} = C_\rho^{(k)} P' \end{cases} \quad (41)$$

Condition:

$$\sum_k \left\{ \frac{\Omega}{\delta t} r_P^{(k)\circ} C_\rho^{(k)} P'_P + \Delta_P [r^{(k)\circ} C_\rho^{(k)} U^{(k)*} P'] - \Delta_P [r^{(k)\circ} \rho^{(k)\circ} (r^{(k)\circ} \mathbf{D}^{(k)} \nabla P') \mathbf{S}] \right\}$$

$$\therefore = - \sum_k \left\{ \begin{array}{l} \frac{r_P^{(k)\circ} \rho_P^{(k)\circ} - (r_P^{(k)} \rho_P^{(k)})^{Old}}{\delta t} \Omega + \Delta_P [r^{(k)\circ} \rho^{(k)\circ} U^{(k)*}] \\ + \Delta_P [r^{(k)\circ} \rho^{(k)\circ} (\mathbf{HP}[\mathbf{u}^{(k)'}]) \mathbf{S}] + \Delta_P [r^{(k)\circ} \rho^{(k)'} \mathbf{u}^{(k)'} \cdot \mathbf{S}] \end{array} \right\} \quad (42)$$

Approximation:

Neglect:

$$\mathbf{HP}[\mathbf{u}^{(k)'}], \Delta_P [r^{(k)\circ} \rho^{(k)'} \mathbf{u}^{(k)'} \cdot \mathbf{S}] \Rightarrow \mathbf{u}_P^{(k)'} = -r^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_P P' \quad (43)$$

Approximate Equation:

$$\sum_k \left\{ \frac{\Omega}{\delta t} r_P^{(k)\circ} C_\rho^{(k)} P'_P + \Delta_P [r^{(k)\circ} C_\rho^{(k)} U^{(k)*} P'] - \Delta_P [r^{(k)\circ} \rho^{(k)\circ} (r^{(k)\circ} \mathbf{D}^{(k)} \nabla P') \mathbf{S}] \right\}$$

$$\Rightarrow = - \sum_k \left\{ \frac{r_P^{(k)\circ} \rho_P^{(k)\circ} - (r_P^{(k)} \rho_P^{(k)})^{Old}}{\delta t} \Omega + \Delta_P [r^{(k)\circ} \rho^{(k)\circ} U^{(k)*}] \right\} \quad (44)$$

A Global MCBA-SIMPLE Iteration

- Solve implicitly for $\mathbf{u}^{(k)}$, using the old pressure and density fields.
- Calculate the $\mathbf{D}^{(k)}$ fields.
- Solve the pressure correction equation.
- Correct $\mathbf{u}^{(k)}$, P and $\rho^{(k)}$.
- Solve implicitly for $r^{(k)}$.
- Solve implicitly the energy equations and update the density fields.
- Return to the first step and iterate until convergence.

The MCBA following SIMPLEC (MCBA-SIMPLEC): Symbolic Form

Predictor:

$$\mathbf{u}_P^{(k)*} = \mathbf{HP}_P[\mathbf{u}^{(k)*}] - r^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_p P^\circ \quad (45)$$

Corrector:

$$(\mathbf{u}^{(k)'}, P', \rho^{(k)'}) (\mathbf{u}^{(k)**} = \mathbf{u}^{(k)*} + \mathbf{u}^{(k)'}, P^* = P^\circ + P', \rho^{(k)*} = \rho^{(k)\circ} + \rho^{(k)'}) \quad (46)$$

$$\therefore \mathbf{u}_P^{(k)**} = \mathbf{HP}_P[\mathbf{u}^{(k)**}] - r^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_p P^* = \mathbf{HP}_P[\mathbf{u}^{(k)*} + \mathbf{u}^{(k)'}] - r^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_p (P^\circ + P') \quad (47)$$

$$\therefore \mathbf{u}_P^{(k)'} = \mathbf{HP}_P[\mathbf{u}^{(k)'}] - r^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_p P' \quad (48)$$

Subtracting $\tilde{\mathbf{HP}}_P[1]\mathbf{u}_P^{(k)'}$ from both sides, one gets

$$\mathbf{u}_P^{(k)'} - \tilde{\mathbf{HP}}_P[1]\mathbf{u}_P^{(k)'} = \mathbf{HP}_P[\mathbf{u}^{(k)'}] - \tilde{\mathbf{HP}}_P[1]\mathbf{u}_P^{(k)'} - r^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_p P' \quad (49)$$

$$\therefore \begin{cases} \mathbf{u}_P^{(k)'} = \frac{\mathbf{HP}_P[\mathbf{u}^{(k)'} - \mathbf{u}_P^{(k)'}]}{1 - \tilde{\mathbf{HP}}_P[1]} - \frac{r^{(k)\circ} \mathbf{D}_P^{(k)}}{1 - \tilde{\mathbf{HP}}_P[1]} \nabla_p P' \\ \rho^{(k)'} = C_\rho^{(k)} P' \end{cases} \quad (50)$$

Condition:

$$\sum_k \left\{ \frac{\Omega}{\delta t} r_P^{(k)\circ} C_\rho^{(k)} P'_P + \Delta_P \left[r^{(k)\circ} C_\rho^{(k)} U^{(k)*} P' \right] - \Delta_P \left[r^{(k)\circ} \rho^{(k)\circ} \left(r^{(k)\circ} \frac{\mathbf{D}^{(k)}}{1 - \tilde{\mathbf{HP}}[1]} \nabla P' \right) \mathbf{S} \right] \right\} \\ = - \sum_k \left\{ \frac{r_P^{(k)\circ} \rho_P^{(k)\circ} - (r_P^{(k)\circ} \rho_P^{(k)})^{Old}}{\delta t} \Omega + \Delta_P \left[r^{(k)\circ} \rho^{(k)\circ} U^{(k)*} \right] \right. \\ \left. + \Delta_P \left[r^{(k)\circ} \rho^{(k)\circ} \left(\frac{\mathbf{HP}[\mathbf{u}^{(k)'} - \mathbf{u}_P^{(k)'}]}{1 - \tilde{\mathbf{HP}}[1]} \right) \mathbf{S} \right] + \Delta_P \left[r^{(k)\circ} \rho^{(k)\circ} \mathbf{u}^{(k)'} \cdot \mathbf{S} \right] \right\} \quad (51)$$

Approximation:

Neglect:

$$\mathbf{HP}[\mathbf{u}^{(k)'} - \mathbf{u}_P^{(k)'}], \Delta_P[r^{(k)\circ} \rho^{(k)\circ} \mathbf{u}^{(k)'} \cdot \mathbf{S}] \Rightarrow \mathbf{u}_P^{(k)'} = -r^{(k)\circ} \tilde{\mathbf{D}}_P^{(k)} \nabla_p P' \quad (52)$$

Approximate Equation:

$$\sum_k \left\{ \frac{\Omega}{\delta t} r_P^{(k)\circ} C_\rho^{(k)} P'_P + \Delta_P \left[r^{(k)\circ} C_\rho^{(k)} U^{(k)*} P' \right] - \Delta_P \left[r^{(k)\circ} \rho^{(k)\circ} (r^{(k)\circ} \tilde{\mathbf{D}}^{(k)} \nabla P') \mathbf{S} \right] \right\} \\ \Rightarrow - \sum_k \left\{ \frac{r_P^{(k)\circ} \rho_P^{(k)\circ} - (r_P^{(k)\circ} \rho_P^{(k)})^{Old}}{\delta t} \Omega + \Delta_P \left[r^{(k)\circ} \rho^{(k)\circ} U^{(k)*} \right] \right\} \quad (53)$$

A Global MCBA-SIMPLEC Iteration

-
- Solve implicitly for $u^{(k)}$, using the old pressure and density fields.
 - Calculate the $\tilde{\mathbf{D}}^{(k)}$ fields.
 - Solve the pressure correction equation.
 - Correct $u^{(k)}$, P and $\rho^{(k)}$.
 - Solve implicitly for $r^{(k)}$.
 - Solve implicitly the energy equations and update the density fields.
 - Return to the first step and iterate until convergence.
-

The MCBA following PRIME (MCBA-PRIME): Symbolic Form

Predictor:

$$\mathbf{u}_P^{(k)*} = \mathbf{H}\mathbf{P}_P[\mathbf{u}^{(k)\circ}] - r^{(k)\circ}\mathbf{D}_P^{(k)}\nabla_P P^\circ \quad (54)$$

Corrector:

$$(\mathbf{u}^{(k)'}, P', \rho^{(k)'}) (\mathbf{u}^{(k)**} = \mathbf{u}^{(k)*} + \mathbf{u}^{(k)'}, P^* = P^\circ + P', \rho^{(k)*} = \rho^{(k)\circ} + \rho^{(k)'}) \quad (55)$$

$$\therefore \mathbf{u}_P^{(k)**} = \mathbf{H}\mathbf{P}_P[\mathbf{u}^{(k)**}] - r^{(k)\circ}\mathbf{D}_P^{(k)}\nabla_P P^* = \mathbf{H}\mathbf{P}_P[\mathbf{u}^{(k)*} + \mathbf{u}^{(k)'}] - r^{(k)\circ}\mathbf{D}_P^{(k)}\nabla_P(P^\circ + P') \quad (56)$$

$$\therefore \begin{cases} \mathbf{u}_P^{(k)'} = \mathbf{H}\mathbf{P}_P[\mathbf{u}^{(k)*} - \mathbf{u}^{(k)\circ}] + \mathbf{H}\mathbf{P}_P[\mathbf{u}^{(k)'}] - r^{(k)\circ}\mathbf{D}_P^{(k)}\nabla_P P' \\ \rho^{(k)'} = C_\rho^{(k)} P' \end{cases} \quad (57)$$

Condition:

$$\sum_k \left\{ \frac{\Omega}{\delta t} r_P^{(k)\circ} C_\rho^{(k)} P'_P + \Delta_P [r^{(k)\circ} C_\rho^{(k)} U^{(k)*} P'] - \Delta_P [r^{(k)} \rho^{(k)\circ} (r^{(k)\circ} \mathbf{D}^{(k)} \nabla P') \mathbf{S}] \right\}$$

$$= - \sum_k \left\{ \begin{array}{l} \left[\frac{r_P^{(k)\circ} \rho_P^{(k)\circ} - (r_P^{(k)} \rho_P^{(k)})^{Old}}{\delta t} \Omega + \Delta_P [r^{(k)\circ} \rho^{(k)\circ} U^{(k)*}] \right] \\ + \Delta_P [r^{(k)\circ} \rho^{(k)\circ} (\mathbf{H}\mathbf{P}[\mathbf{u}^{(k)*} - \mathbf{u}^{(k)}] + \mathbf{H}\mathbf{P}[\mathbf{u}^{(k)}]) \mathbf{S}] + \Delta_P [r^{(k)} \rho^{(k)'} \mathbf{u}^{(k)'} \cdot \mathbf{S}] \end{array} \right\} \quad (58)$$

Approximation:

$$\text{Neglect: } \mathbf{H}\mathbf{P}[\mathbf{u}^{(k)*} - \mathbf{u}^{(k)\circ}], \mathbf{H}\mathbf{P}[\mathbf{u}^{(k)}], \Delta_P [r^{(k)\circ} \rho^{(k)'} \mathbf{u}^{(k)} \cdot \mathbf{S}]$$

Approximate Equation:

$$\sum_k \left\{ \frac{\Omega}{\delta t} r_P^{(k)\circ} C_\rho^{(k)} P'_P + \Delta_P [r^{(k)\circ} C_\rho^{(k)} U^{(k)*} P'] - \Delta_P [r^{(k)} \rho^{(k)\circ} (r^{(k)\circ} \mathbf{D}^{(k)} \nabla P') \mathbf{S}] \right\}$$

$$= - \sum_k \left\{ \left[\frac{r_P^{(k)\circ} \rho_P^{(k)\circ} - (r_P^{(k)} \rho_P^{(k)})^{Old}}{\delta t} \Omega + \Delta_P [r^{(k)\circ} \rho^{(k)\circ} U^{(k)*}] \right] \right\} \quad (59)$$

A Global MCBA-PRIME Iteration

-
- Solve explicitly for $u^{(k)}$, using the old pressure and density fields.
 - Calculate the $\mathbf{D}^{(k)}$ fields.
 - Solve the pressure correction equation.
 - Correct $u^{(k)}$, P and $\rho^{(k)}$.
 - Solve implicitly for $r^{(k)}$.
 - Solve implicitly the energy equations and update the density fields.
 - Return to the first step and iterate until convergence.
-

The MCBA following SIMPLEST (MCBA-SIMPLEST): Symbolic Form

Predictor:

$$\mathbf{u}_P^{(k)*} = \mathbf{HP}_P^D [\mathbf{u}^{(k)*}] + \mathbf{HP}_P^C [\mathbf{u}^{(k)\circ}] - r^{(k)\circ} \mathbf{D}_P^{(k)*} \nabla_p P^\circ \quad (60)$$

Corrector:

$$(\mathbf{u}^{(k)'}, P', \rho^{(k)'}) (\mathbf{u}^{(k)**} = \mathbf{u}^{(k)*} + \mathbf{u}^{(k)'}, P^* = P^\circ + P', \rho^{(k)*} = \rho^{(k)\circ} + \rho^{(k)'}) \quad (61)$$

$$\begin{aligned} \mathbf{u}_P^{(k)**} &= \mathbf{HP}_P^D [\mathbf{u}^{(k)*}] + \mathbf{HP}_P^D [\mathbf{u}^{(k)'}] + \mathbf{HP}_P^C [\mathbf{u}^{(k)*}] + \mathbf{HP}_P^C [\mathbf{u}^{(k)'}] \\ &\quad - r^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_p P^\circ - r^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_p P' \end{aligned} \quad (62)$$

$$\begin{cases} \mathbf{u}_P^{(k)'} = \mathbf{HP}_P [\mathbf{u}^{(k)'}] + \mathbf{HP}_P^C [\mathbf{u}^{(k)*} - \mathbf{u}^{(k)\circ}] - r^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_p P' \\ \rho^{(k)'} = C_\rho^{(k)} P' \end{cases} \quad (63)$$

Condition:

$$\begin{aligned} \sum_k \left\{ \frac{\Omega}{\delta t} r_P^{(k)\circ} C_\rho^{(k)} P'_P + \Delta_P [r^{(k)\circ} C_\rho^{(k)} U^{(k)*} P'] - \Delta_P [r^{(k)\circ} \rho^{(k)\circ} (r^{(k)\circ} \mathbf{D}^{(k)} \nabla P') \mathbf{S}] \right. \\ \left. = - \sum_k \left\{ \frac{r_P^{(k)\circ} \rho_P^{(k)\circ} - (r_P^{(k)} \rho_P^{(k)})^{Old}}{\delta t} \Omega + \Delta_P [r^{(k)\circ} \rho^{(k)\circ} U^{(k)*}] \right\} \right. \\ \left. + \Delta_P [r^{(k)\circ} \rho^{(k)\circ} (\mathbf{HP}^C [\mathbf{u}^{(k)*} - \mathbf{u}^{(k)\circ}] + \mathbf{HP} [\mathbf{u}^{(k)'}]) \mathbf{S}] + \Delta_P [r^{(k)\circ} \rho^{(k)'} \mathbf{u}^{(k)'} \cdot \mathbf{S}] \right] \end{aligned} \quad (64)$$

Approximation:

Neglect:

$$\mathbf{HP}^C [\mathbf{u}^{(k)*} - \mathbf{u}^{(k)\circ}], \mathbf{HP} [\mathbf{u}^{(k)'}], \Delta_P [r^{(k)\circ} \rho^{(k)'} \mathbf{u}^{(k)'} \cdot \mathbf{S}] \Rightarrow \mathbf{u}_P^{(k)'} = -r^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_p P' \quad (65)$$

Approximate Equation:

$$\begin{aligned} \sum_k \left\{ \frac{\Omega}{\delta t} r_P^{(k)\circ} C_\rho^{(k)} P'_P + \Delta_P [r^{(k)\circ} C_\rho^{(k)} U^{(k)*} P'] - \Delta_P [r^{(k)\circ} \rho^{(k)\circ} (r^{(k)\circ} \mathbf{D}^{(k)} \nabla P') \mathbf{S}] \right\} \\ = - \sum_k \left\{ \frac{r_P^{(k)\circ} \rho_P^{(k)\circ} - (r_P^{(k)} \rho_P^{(k)})^{Old}}{\delta t} \Omega + \Delta_P [r^{(k)\circ} \rho^{(k)\circ} U^{(k)*}] \right\} \end{aligned} \quad (66)$$

A Global MCBA-SIMPLEST Iteration

-
- Solve for $\mathbf{u}^{(k)}$, treating diffusion implicitly and convection explicitly.
 - Calculate the $\mathbf{D}^{(k)}$ fields.
 - Solve the pressure correction equation.
 - Correct $\mathbf{u}^{(k)}$, P and $\rho^{(k)}$.
 - Solve implicitly for $r^{(k)}$.
 - Solve implicitly the energy equations and update the density fields.
 - Return to the first step and iterate until convergence.
-

The MCBA following PISO (MCBA-PISO): Symbolic Form

First Predictor:

$$\mathbf{u}_P^{(k)*} = \mathbf{HP}_P [\mathbf{u}^{(k)*}] - r^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_p P^\circ \quad (67)$$

First Corrector:

$$(\mathbf{u}^{(k)'}, P', \rho^{(k)'}) (\mathbf{u}^{(k)**} = \mathbf{u}^{(k)*} + \mathbf{u}^{(k)'}, P^* = P^\circ + P', \rho^{(k)*} = \rho^{(k)\circ} + \rho^{(k)'}) \quad (68)$$

$$\therefore \mathbf{u}_P^{(k)**} = \mathbf{H}\mathbf{P}_P[\mathbf{u}^{(k)**}] - r^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_P P^* = \mathbf{H}\mathbf{P}_P[\mathbf{u}^{(k)*} + \mathbf{u}^{(k)'}] - r^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_P (P^\circ + P') \quad (69)$$

$$\therefore \begin{cases} \mathbf{u}_P^{(k)'} = \mathbf{H}\mathbf{P}_P[\mathbf{u}^{(k)'}] - r^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_P P' \\ \rho^{(k)'} = C_\rho^{(k)} P' \end{cases} \quad (70)$$

Condition:

$$\sum_k \left\{ \frac{\Omega}{\delta t} r_P^{(k)\circ} C_\rho^{(k)} P'_P + \Delta_P [r^{(k)\circ} C_\rho^{(k)} U^{(k)*} P'] - \Delta_P [r^{(k)\circ} \rho^{(k)\circ} (r^{(k)\circ} \mathbf{D}^{(k)} \nabla P') \mathbf{S}] \right\} = - \sum_k \left\{ \begin{array}{l} \frac{r_P^{(k)\circ} \rho_P^{(k)\circ} - (r_P^{(k)} \rho_P^{(k)})^{Old}}{\delta t} \Omega + \Delta_P [r^{(k)\circ} \rho^{(k)\circ} U^{(k)*}] \\ + \Delta_P [r^{(k)\circ} \rho^{(k)\circ} (\mathbf{H}\mathbf{P}[\mathbf{u}^{(k)'}]) \mathbf{S}] + \Delta_P [r^{(k)\circ} \rho^{(k)'} \mathbf{u}^{(k)'} \cdot \mathbf{S}] \end{array} \right\} \quad (71)$$

Approximation:

Neglect:

$$\mathbf{H}\mathbf{P}[\mathbf{u}^{(k)'}], \Delta_P [r^{(k)\circ} \rho^{(k)'} \mathbf{u}^{(k)'} \cdot \mathbf{S}] \Rightarrow \mathbf{u}_P^{(k)'} = -r^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_P P' \quad (72)$$

Approximate Equation:

$$\sum_k \left\{ \frac{\Omega}{\delta t} r_P^{(k)\circ} C_\rho^{(k)} P'_P + \Delta_P [r^{(k)\circ} C_\rho^{(k)} U^{(k)*} P'] - \Delta_P [r^{(k)\circ} \rho^{(k)\circ} (r^{(k)\circ} \mathbf{D}^{(k)} \nabla P') \mathbf{S}] \right\} = - \sum_k \left\{ \frac{r_P^{(k)\circ} \rho_P^{(k)\circ} - (r_P^{(k)} \rho_P^{(k)})^{Old}}{\delta t} \Omega + \Delta_P [r^{(k)\circ} \rho^{(k)\circ} U^{(k)*}] \right\} \quad (73)$$

Second Corrector:

$$(\mathbf{u}^{(k)''}, P'', \rho^{(k)''}) (\mathbf{u}^{(k)****} = \mathbf{u}^{(k)***} + \mathbf{u}^{(k)''}, P^{**} = P^* + P'', \rho^{(k)**} = \rho^{(k)*} + \rho^{(k)''}) \quad (74)$$

$$\therefore \mathbf{u}_P^{(k)****} = \mathbf{H}\mathbf{P}_P^{**} [\mathbf{u}^{(k)****}] - r^{(k)\circ} \mathbf{D}_P^{(k)**} \nabla_P (P^* + P'') \quad (75)$$

$$\mathbf{u}_P^{(k)***} = \mathbf{H}\mathbf{P}_P^{**} [\mathbf{u}^{(k)**}] - r^{(k)\circ} \mathbf{D}_P^{(k)**} \nabla_P P^* \quad (76)$$

$$\therefore \begin{cases} \mathbf{u}_P^{(k)''} = \mathbf{H}\mathbf{P}_P^{**} [\mathbf{u}^{(k)***} - \mathbf{u}^{(k)**} + \mathbf{u}^{(k)''}] - r^{(k)\circ} \mathbf{D}_P^{(k)**} \nabla_P P'' \\ \rho^{(k)''} = C_\rho^{(k)} P'' \end{cases} \quad (77)$$

Condition:

$$\sum_k \left\{ \frac{\Omega}{\delta t} r_P^{(k)\circ} C_\rho^{(k)} P''_P + \Delta_P [r^{(k)\circ} C_\rho^{(k)} U^{(k)***} P''] - \Delta_P [r^{(k)\circ} \rho^{(k)*} (r^{(k)\circ} \mathbf{D}^{(k)**} \nabla P'') \mathbf{S}] \right\} = - \sum_k \left\{ \begin{array}{l} \frac{r_P^{(k)\circ} \rho_P^{(k)*} - (r_P^{(k)} \rho_P^{(k)})^{Old}}{\delta t} \Omega + \Delta_P [r^{(k)\circ} \rho^{(k)*} U^{(k)***}] \\ + \Delta_P [r^{(k)\circ} \rho^{(k)*} (\mathbf{H}\mathbf{P}^{**} [\mathbf{u}^{(k)***} - \mathbf{u}^{(k)**} + \mathbf{u}^{(k)''}]) \mathbf{S}] + \Delta_P [r^{(k)\circ} \rho^{(k)''} \mathbf{u}^{(k)''} \cdot \mathbf{S}] \end{array} \right\} \quad (78)$$

Approximation:

Neglect:

$$\mathbf{H}\mathbf{P}^{**} \left[\mathbf{u}^{(k)**} - \mathbf{u}^{(k)**} + \mathbf{u}^{(k)'} \right], \Delta_P \left[r^{(k)\circ} \rho^{(k)''} \mathbf{u}^{(k)'}.S \right] \Rightarrow \mathbf{u}_P^{(k)'} = -r^{(k)\circ} \mathbf{D}_P^{(k)**} \nabla_P P'' \quad (79)$$

Approximate Equation:

$$\begin{aligned} \sum_k \left\{ \frac{\Omega}{\delta t} r_P^{(k)\circ} C_\rho^{(k)} P'' + \Delta_P \left[r^{(k)\circ} C_\rho^{(k)} U^{(k)**} P'' \right] - \Delta_P \left[r^{(k)\circ} \rho^{(k)*} \left(r^{(k)\circ} \mathbf{D}^{(k)**} \nabla_P P'' \right) S \right] \right\} \\ = -\sum_k \left\{ \frac{r_P^{(k)\circ} \rho_P^{(k)*} - (r_P^{(k)} \rho_P^{(k)})^{Old}}{\delta t} \Omega + \Delta_P \left[r^{(k)\circ} \rho^{(k)*} U^{(k)**} \right] \right\} \end{aligned} \quad (80)$$

A Global MCBA-PISO Iteration

-
- Solve implicitly for $\mathbf{u}^{(k)}$ using the old pressure and density fields.
 - Calculate the $\mathbf{D}^{(k)}$ fields.
 - Solve the pressure correction equation.
 - Correct $\mathbf{u}^{(k)}$, P , and $\rho^{(k)}$.
 - Solve implicitly for $r^{(k)}$.
 - Solve implicitly the energy equation and update the density fields.
 - Solve the momentum equations explicitly and calculate the $\mathbf{D}^{(k)}$ fields.
 - Solve the pressure correction equation.
 - Correct $\mathbf{u}^{(k)}$, P , and $\rho^{(k)}$.
 - Return to step one and iterate until convergence
-

The MCBA following SIMPLEX (MCBA-SIMPLEX): Symbolic FormPredictor:

$$\mathbf{u}_P^{(k)*} = \mathbf{H}\mathbf{P}_P [\mathbf{u}^{(k)*}] - r^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_P P^o \quad (81)$$

Corrector:

$$(\mathbf{u}^{(k)'}, P', \rho^{(k)'}) (\mathbf{u}^{(k)**} = \mathbf{u}^{(k)*} + \mathbf{u}^{(k)'}, P^* = P^o + P', \rho^{(k)*} = \rho^{(k)\circ} + \rho^{(k)'}) \quad (82)$$

$$\therefore \mathbf{u}_P^{(k)**} = \mathbf{H}\mathbf{P}_P [\mathbf{u}^{(k)**}] - r^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_P P^* = \mathbf{H}\mathbf{P}_P [\mathbf{u}^{(k)*} + \mathbf{u}^{(k)'}] - r^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_P (P^o + P') \quad (83)$$

$$\therefore \begin{cases} \mathbf{u}_P^{(k)'} = \mathbf{H}\mathbf{P}_P [\mathbf{u}^{(k)'}] - r^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_P P' \\ \rho^{(k)'} = C_\rho^{(k)} P' \end{cases} \quad (84)$$

Condition:

$$\sum_k \left\{ \frac{r_P^{(k)\circ} (\rho_P^{(k)\circ} + \rho_P^{(k)'}) - (r_P^{(k)} \rho_P^{(k)})^{Old}}{\delta t} \Omega \right\} = -\sum_k \left\{ \Delta_P \left[r^{(k)\circ} \rho^{(k)\circ} \mathbf{u}^{(k)*}.S \right] + \Delta_P \left[r^{(k)\circ} \rho^{(k)'} \mathbf{u}^{(k)'} .S \right] \right\} \quad (85)$$

Approximation:

Neglect $\Delta_P [r^{(k)\circ} \rho^{(k)'} \mathbf{u}^{(k)'} .S]$ and let

$$\mathbf{u}_P^{(k)'} = \mathbf{H}\mathbf{P}_P [\mathbf{u}^{(k)'}] - r_P^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_P P' = -r_P^{(k)\circ} \mathbf{D}_P^{(k)SX} \nabla_P P' \quad (86)$$

$$\Rightarrow -r_P^{(k)\circ} \mathbf{D}_P^{(k)SX} \nabla_P P' = \mathbf{H}\mathbf{P}_P [-r^{(k)\circ} \mathbf{D}^{(k)SX} \nabla_P P'] - r_P^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_P P' \quad (87)$$

Assume that the pressure difference local to the velocity is representative of all pressure differences

i.e. $\mathbf{HP}_P[-r^{(k)\circ}\mathbf{D}^{(k)SX}\nabla_P P'] = -(\nabla_P P')\mathbf{HP}_P[r^{(k)\circ}\mathbf{D}^{(k)SX}]$, thus:

$$r_P^{(k)\circ}\mathbf{D}_P^{(k)SX} = \mathbf{HP}_P[r^{(k)\circ}\mathbf{D}^{(k)SX}] + r_P^{(k)\circ}\mathbf{D}_P^{(k)} \quad (88)$$

Approximate Equation:

$$\begin{aligned} \sum_k \left\{ \frac{\Omega}{\delta t} r_P^{(k)\circ} C_\rho^{(k)} P'_P + \Delta_P [r^{(k)\circ} C_\rho^{(k)} U^{(k)*} P'] - \Delta_P \left[r^{(k)\circ} \rho^{(k)\circ} (r^{(k)\circ} \mathbf{D}^{(k)SX} \nabla P') S \right] \right\} \\ = - \sum_k \left\{ \frac{r_P^{(k)\circ} \rho_P^{(k)\circ} - (r_P^{(k)\circ} \rho_P^{(k)})^{Old}}{\delta t} \Omega + \Delta_P [r^{(k)\circ} \rho^{(k)\circ} U^{(k)*}] \right\} \end{aligned} \quad (89)$$

A Global MCBA-SIMPLEX Iteration

-
- Solve implicitly for $\mathbf{u}^{(k)}$, using the old pressure and density fields.
 - Calculate the $\mathbf{D}^{(k)}$ fields.
 - Solve implicitly for the $\mathbf{D}^{(k)SX}$ fields.
 - Solve the pressure correction equation using these $\mathbf{D}^{(k)SX}$ fields.
 - Correct $\mathbf{u}^{(k)}$, P and $\rho^{(k)}$.
 - Solve implicitly for $r^{(k)}$.
 - Solve implicitly the energy equations and update the density fields.
 - Return to the first step and iterate until convergence.
-

The MCBA following SIMPLEM (MCBA-SIMPLEM): Symbolic Form

First Predictor:

No predictor stage. Only coefficients of the momentum equations are calculated.

First Corrector:

$$(\mathbf{u}^{(k)'}, P', \rho^{(k)'}) (\mathbf{u}^{(k)*} = \mathbf{u}^{(k)\circ} + \mathbf{u}^{(k)'}, P^* = P^\circ + P', \rho^{(k)*} = \rho^{(k)\circ} + \rho^{(k)'}) \quad (90)$$

$$\therefore \mathbf{u}_P^{(k)*} = \mathbf{HP}_P[\mathbf{u}^{(k)*}] - r^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_P P^* \quad (91)$$

$$\mathbf{u}_P^{(k)\circ} = \mathbf{HP}_P[\mathbf{u}^{(k)\circ}] - r^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_P P^\circ \quad (92)$$

$$\therefore \begin{cases} \mathbf{u}_P^{(k)\circ} = \mathbf{HP}_P[\mathbf{u}^{(k)\circ}] - r^{(k)\circ} \mathbf{D}_P^{(k)} \nabla_P P' \\ \rho^{(k)\circ} = C_\rho^{(k)} P' \end{cases} \quad (93)$$

Condition:

$$\begin{aligned} \sum_k \left\{ \frac{\Omega}{\delta t} r_P^{(k)\circ} C_\rho^{(k)} P'_P + \Delta_P [r^{(k)\circ} C_\rho^{(k)} U^{(k)\circ} P'] - \Delta_P \left[r^{(k)\circ} \rho^{(k)\circ} (r^{(k)\circ} \mathbf{D}^{(k)} \nabla P') S \right] \right\} \\ \therefore = - \sum_k \left\{ \frac{r_P^{(k)\circ} \rho_P^{(k)\circ} - (r_P^{(k)\circ} \rho_P^{(k)})^{Old}}{\delta t} \Omega + \Delta_P [r^{(k)\circ} \rho^{(k)\circ} U^{(k)\circ}] \right. \\ \left. + \Delta_P [r^{(k)\circ} \rho^{(k)\circ} (\mathbf{HP}[\mathbf{u}^{(k)\circ}]) S] + \Delta_P [r^{(k)\circ} \rho^{(k)\circ} \mathbf{u}^{(k)\circ} \cdot \mathbf{S}] \right\} \end{aligned} \quad (94)$$

Approximation:

Neglect:

$$\mathbf{H}\mathbf{P}[\mathbf{u}^{(k)'}], \Delta_p[r^{(k)} \rho^{(k)} \mathbf{u}^{(k)'} \cdot \mathbf{s}] \Rightarrow \mathbf{u}_p^{(k)'} = -r^{(k)} \mathbf{D}_p^{(k)} \nabla_p P' \quad (95)$$

Approximate Equation:

$$\begin{aligned} \sum_k \left\{ \frac{\Omega}{\delta t} r_p^{(k)} C_p^{(k)} P'_p + \Delta_p [r^{(k)} C_p^{(k)} U^{(k)} P'] - \Delta_p [r^{(k)} \rho^{(k)} (r^{(k)} \mathbf{D}^{(k)} \nabla P') \mathbf{s}] \right\} \\ = - \sum_k \left\{ \frac{r_p^{(k)} \rho_p^{(k)} - (r_p^{(k)} \rho_p^{(k)})^{old}}{\delta t} \Omega + \Delta_p [r^{(k)} \rho^{(k)} U^{(k)}] \right\} \end{aligned}$$

Second Predictor:

$$\mathbf{u}_p^{(k)*} = \mathbf{H}\mathbf{P}_p[\mathbf{u}^{(k)*}] - r^{(k)} \mathbf{D}_p^{(k)*} \nabla_p P^* \quad (96)$$

Second Corrector:

No corrector stage.

A Global MCBA-SIMPLEM Iteration

-
- Calculate the $\mathbf{D}^{(k)}$ fields based on values from the previous iteration.
 - Solve the pressure correction equation.
 - Correct $\mathbf{u}^{(k)}$, P and $\rho^{(k)}$.
 - Calculate new $\mathbf{H}\mathbf{P}^{(k)}$ and $\mathbf{D}^{(k)}$ fields.
 - Solve implicitly for $\mathbf{u}^{(k)}$ using the new fields.
 - Solve implicitly for $r^{(k)}$.
 - Solve implicitly the energy equations and update the density fields.
 - Return to the first step and iterate until convergence.
-

The Expanded Form of the Pressure-Correction Equation

In simplified form, the pressure-correction equation may be written as:

$$\begin{aligned} \sum_k \left\{ \frac{\Omega}{\delta t} r_p^{(k)} C_p^{(k)} P'_p + \Delta_p [r^{(k)} C_p^{(k)} U^{(k)} P'] - \Delta_p [r^{(k)} \rho^{(k)} (r^{(k)} \mathbf{D}^{(k)} \nabla P') \mathbf{s}] \right\} \\ = - \sum_k \left\{ \frac{r_p^{(k)} \rho_p^{(k)} - (r_p^{(k)} \rho_p^{(k)})^{old}}{\delta t} \Omega + \Delta_p [r^{(k)} \rho^{(k)} U^{(k)}] \right\} \quad (97) \end{aligned}$$

Where, depending on the algorithm used, $U^{(k)}$ and $\rho^{(k)}$ represent values from the previous iteration or from a previous corrector step. The second term on the left hand side is similar to a convection term and may be discretized using any convective scheme (the upwind scheme is employed here). The term $\Delta_p [r^{(k)} \rho^{(k)} (r^{(k)} \mathbf{D}^{(k)} \nabla P') \mathbf{s}]$ is discretized following the same procedure used in discretizing the diffusion flux. Substituting the various terms in Eq. (97) by their equivalent expressions and neglecting the non-orthogonal terms, the final form of the pressure-correction equation is written as:

$$A_p^{P'} P'_p = \sum_{nb} A_{nb}^{P'} P'_{nb} + B_p^{P'} \quad (98)$$

where

$$A_F^{P'} = \sum_k \left(\Gamma_f^{(k)} + \left(r^{(k)\circ} C_\rho^{(k)} \right)_f \| -U_f^{(k)}, 0 \| \right) \quad (99)$$

$$A_P^{P'} = \sum_{NB} A_F^{P'} + \sum_k \left(\frac{\left(r^{(k)\circ} C_\rho^{(k)} \right)_P \Omega_P}{\delta t} + \sum_{nb} \left(r^{(k)\circ} C_\rho^{(k)} \right)_f U_f^{(k)} \right) \quad (100)$$

$$B_P^{P'} = - \sum_k \left\{ \frac{\left(r_p^{(k)\circ} \rho_p^{(k)} - r_p^{(k)\text{old}} \rho_p^{(k)\text{old}} \right) \Omega_P}{\delta t} + \sum_{nb} r_f^{(k)} \rho_f^{(k)} U_f^{(k)} \right\} \quad (101)$$

$$\Gamma_f = r_f^{(k)\circ} r_f^{(k)\circ} \rho_f^{(k)} \frac{\sum_i \bar{D}_f^{(k)} [u_i] (S_f^{x_i})^2}{S_f \cdot d_f}$$

The corrections are then applied to the velocity, pressure, and density fields using the following equations:

$$u_p^{(k)*} = u_p^{(k)\circ} - r^{(k)\circ} D_p^{(k)} \nabla_p P', P^* = P^\circ + P', \rho^{(k)*} = \rho^{(k)\circ} + C_\rho^{(k)} P' \quad (102)$$

As detailed in [59], numerical experiments using the above approach to simulate air-water flows have shown poor conservation of the lighter fluid. Without going into details, this problem can be considerably alleviated by normalizing the individual continuity equations, and hence the global mass conservation equation, by means of a weighting factor such as a reference density $\rho^{(k)}$ (which is fluid dependent). This approach has been adopted in solving all problems presented in this work.

Solving the energy equations

The solutions of the energy equations follow that of the general multi-fluid scalar equation. As such, nothing new needs to be added in that regard (though in many cases coupling of the energy equation with the momentum and continuity equations is beneficial).

Results and Discussion

The performance of the various multi-fluid Mass Conservation Based Algorithms is assessed in this section by presenting solutions to several one and two-dimensional two-phase problems. Results are presented in terms of the CPU-time needed to converge the solution to a set level and of the convergence history. Moreover, solutions are obtained for a number of grids in order to assess the performance of the various algorithms with increasing grid density. For the two-dimensional

problems, in addition to the CPU-time needed to solve a problem using the single grid method (SG), the CPU-time needed using different solution strategies is also displayed. These include the prolongation scheme (PG) in which the solution from the next coarser grid is used to provide the initial field, and the full multi-grid method (FMG). Results are compared against available experimental data and/or numerical/theoretical values. The residual of a variable ϕ at the end of an outer iteration is defined as:

$$\text{RES}_{\phi}^{(k)} = \sum_{c.v.} \left| A_p \phi_p^{(k)} - \sum_{\text{all } p \text{ neighbours}} A_{nb} \phi_{nb}^{(k)} - B_p^{(k)} \right| \quad (103)$$

For global mass conservation, the imbalance in mass is defined as:

$$\text{RES}_C = \sum_k \sum_{c.v.} \left| \frac{(r_p^{(k)} \rho_p^{(k)}) - (r_p^{(k)} \rho_p^{(k)})^{\text{Old}}}{\delta t} \Omega - \Delta_p [r^{(k)} \rho^{(k)} \mathbf{u}^{(k)} \cdot \mathbf{S}] - r^{(k)} \dot{M}^{(k)} \right| \quad (104)$$

All residuals are normalized by their respective inlet fluxes. Computations are terminated when the maximum normalized residual of all variables, drops below a very small number ϵ_s . For a given problem, the same value of ϵ_s is used with all algorithms. In general, it is found that requiring the overall mass residuals to be satisfied to within ϵ_s is a very stringent requirement and the last to be fulfilled. This is why these residuals are the ones presented here and used to compare the performance of the various algorithms. In all problems, the first phase represents the continuous phase (denoted by a superscript (c)), which must be fluid, and the second phase is the disperse phase (denoted by a superscript (d)), which may be solid or fluid. Unless otherwise specified the HR SMART scheme is used in all computations reported in this study. For a given problem, all results are generated starting from the same initial guess. Moreover, it should be stated that in iterative techniques, different initial guesses might require different computational efforts.

One-dimensional two-phase validation problems

Due to the large number of parameters affecting the performance of the various algorithms and to allow a thorough testing of these algorithms, eight one-dimensional two-phase problems are considered. These problems can be broadly classified as: (i) horizontal particle transport, and (ii) vertical particle transport. Despite its geometric simplicity, the one dimensional particle transport

problem represents a wide range of physical conditions. The effects of grid refinement on accuracy and convergence are studied by solving the problems on four grid systems of sizes 20, 40, 80, and 160 control volumes with ϵ_s assigned the value of 10^{-8} .

Many runs were performed so as to set the control parameters of each algorithm near optimum values. The task was performed once and was not repeated for the 2-dimensional problems. The CPU times are reported in the form of charts and not via numerical values. This allows a comparative assessment of their performance. Moreover, all CPU times are normalized by the time needed by MCBA-SIMPLE to reach the set residuals on the coarsest grid.

Horizontal particle transport

The physical situation is depicted in Fig. 2. Depending on the set densities, it represents either the steady flow of solid particles suspended in a free stream of air or the steady flow of air bubbles in a stream of water. The slip between the phases determines the drag, which is the sole driving force for the particle-bubble/air-water motion ($g=0$). In the suspension, the inter-particle/bubble forces are neglected. Diffusion in both phases is set to zero and the inter-phase drag force is calculated as:

$$I_M^{(c)} = \frac{3}{8} \frac{C_D}{r_p} r^{(d)} \rho^{(c)} V_{\text{slip}} (u^{(d)} - u^{(c)}) \quad (105)$$

$$I_M^{(d)} = -\frac{3}{8} \frac{C_D}{r_p} r^{(d)} \rho^{(c)} V_{\text{slip}} (u^{(d)} - u^{(c)}) \quad (106)$$

$$V_{\text{slip}} = \|u^{(d)} - u^{(c)}\| \quad (107)$$

The drag coefficient, C_D , is set to 0.44. Since diffusion is neglected, the MCBA-SIMPLEST and MCBA-PRIME becomes identical and reference will be made to MCBA-SIMPLEST only. The task is to calculate the particle/bubble-velocity distribution as a function of position. If the flow field is extended far enough (here computations are performed over a length of $L=2m$), the particle/bubble and fluid phases are expected to approach an equilibrium velocity given by:

$$U_{\text{equilibrium}} = r_{\text{inlet}}^{(c)} V_{\text{inlet}}^{(c)} + r_{\text{inlet}}^{(d)} V_{\text{inlet}}^{(d)} \quad (108)$$

Dilute gas-solid flow

The steady flow of dilute particles suspended in a free stream of air is studied first. At inlet, the air and particle velocities are 5 m/s and 1 m/s, respectively. The physical properties of the two phases

are: $\rho^{(d)} / \rho^{(c)} = 2000$, $\vartheta^{(c)} = 10^{-5}$, $r_p = 1 \text{ mm}$, $r_{\text{inlet}}^{(d)} = 10^{-5}$. Due to the dilute concentration of the particles, the free stream velocity is more or less unaffected by their presence and the equilibrium velocity is nearly equal to the inlet free stream velocity. Based on this observation, Morsi and Alexander [80] obtained the following analytical solution for the particle velocity $u^{(d)}$ as a function of the position x and the properties of the two phases:

$$\ln[V_{\text{inlet}}^{(c)} - u^{(d)}] + \frac{V_{\text{inlet}}^{(c)}}{V_{\text{inlet}}^{(c)} - u^{(d)}} = \frac{3}{8} \frac{\rho^{(c)}}{\rho^{(d)}} \frac{C_D}{r_p} x + \ln[V_{\text{inlet}}^{(c)} - V_{\text{inlet}}^{(d)}] + \frac{V_{\text{inlet}}^{(c)}}{V_{\text{inlet}}^{(c)} - V_{\text{inlet}}^{(d)}} \quad (109)$$

This case is of particular importance since the flow situation has an exact solution. As shown in Fig. 3(a) the predicted particle velocity distribution falls on top of the analytical solution given by Eq. (109), which is an indication of the accuracy of the numerical procedure. The convergence history of the various MCBA and over the four grid networks used are displayed in Figs. 3(b)-3(h). For all algorithms, the required number of iterations increases as the grid size is increased, with PISO (Fig. 3(b)) requiring the minimum and SIMPLEST/PRIME (Fig. 3(f)) the maximum number of iterations on all grids. The convergence histories of SIMPLE, SIMPLEC, and SIMPLEX (Figs. 3(c), 3(d), and 3(g), respectively) are very similar requiring nearly the same number of iterations on all grids. The convergence path of SIMPLEM (Fig. 3(e)) is not as smooth as that of SIMPLE due to the fact that at the end of an outer iteration, the velocity field is momentum satisfying rather than continuity satisfying. The convergence paths of the various algorithms over a grid of size 80 C.V. are compared in Fig. 3(h) and the above observations can easily be inferred from the figure. The normalized CPU efforts required by the various algorithms over all grids are depicted in Fig. 4. As expected, the chart clearly shows that the CPU time increases with increasing grid density. Moreover, it is hard to see any noticeable difference in the CPU times for SIMPLE, SIMPLEC, SIMPLEX, and PISO. The SIMPLEM algorithm requires slightly higher computational effort as compared to SIMPLE. The worst performance is for SIMPLEST which degenerates to PRIME in the absence of diffusion and results in a fully explicit solution scheme.

Dense gas-solid flow

The only difference between this case and the previous one is in the concentration of particles, which is set to $r_{inlet}^{(d)} = 10^{-2}$. Despite the low value of the inlet disperse phase volume fraction, this problem is dense in the sense that the ratio of disperse phase and continuous phase mass loadings is large $r^{(d)}\rho^{(d)}/r^{(c)}\rho^{(c)} = 20$. Thus the disperse phase carries most of the inertia of the mixture. The equilibrium velocity in this case, as obtained from Eq. (108) is 4.96 m/s as compared to 4.99996 m/s in the previous case. Due to this slight difference between the inlet air velocity and the final equilibrium velocity, the free stream velocity may be assumed to be nearly constant and the variation in particle velocity can be obtained from Eq. (109) (i.e. the variation in particle velocity is exactly the same as in the previous case). The predicted air and particle velocity distributions are displayed in Fig. 5(a). The numerical and analytical particle velocity profiles are indistinguishable and fall on top of each others (denoted solid in the Figure). Moreover, the slight decrease in the air velocity can be easily depicted. The convergence paths for all algorithms and over all grid systems used are displayed in Figs. 5(b)-5(h). In general, a larger number of iterations are required to reach the desired level of convergence on a given grid as compared to the dilute case due to the increased importance of the inter-phase term. The general convergence trend is similar to that of the dilute problem with PISO requiring the minimum and SIMPLEST the maximum number of iterations. The SIMPLEX algorithm (Fig. 5(g)) is seen to require a slightly lower number of iterations on the finest grid as compared to SIMPLE (Fig. 5(c)), SIMPLEC (Fig. 5(d)), and SIMPLEM (Fig. 5(e)). The smoothest convergence paths are for SIMPLEX and SIMPLEC. As depicted in Figs. 5(f) and 5(h), the performance of SIMPLEST/PRIME is poor as compared to other algorithms for the same reasons stated above. The normalized CPU times required by the various algorithms on all grids are presented in Fig. 6. The same conclusions drawn above are applicable here with PISO requiring the lowest computational effort (10% less than SIMPLE on the finest mesh). The performance of SIMPLE, SIMPLEC, and SIMPLEX is more or less identical while that of SIMPLEM is of lower quality necessitating increasingly higher computational effort with denser meshes, and requiring about 40% more effort than SIMPLE on the fine grids (80 and 160 C.V.). The computational effort needed by

SIMPLEST/PRIME is however the most extensive and is nearly 623% the one needed by SIMPLE on the finest mesh.

Dilute bubbly flow

For the same configuration displayed in Fig. 2, the continuous phase is considered to be water and the disperse phase to be air. The resulting flow is denoted in the literature by bubbly flow. With the exception of $\rho^{(d)} / \rho^{(c)} = 10^{-3}$ and at inlet $r_{inlet}^{(d)} = 0.1$, other physical properties and inlet conditions are the same as those considered earlier. This is a strongly coupled problem and represents a good test for the numerical procedure and performance of the algorithms. The correct physical solution is that the bubble and continuous phase velocities both reach the equilibrium velocity of 4.6 m/s (Eq. (108)) in a distance too small to be correctly resolved by any of the grid networks used. Results for this case are presented in Figs. 7 and 8. Axial velocity distribution for both water and air are displayed in Fig. 7(a). Both phases reach the equilibrium velocity of 4.6 m/s over a very short distance from the inlet section and remain constant afterward, as expected. The relative convergence characteristics of the various algorithms remain the same. However, all algorithms require larger number of iterations as compared to the dilute gas solid flow case due to the stronger coupling between the phases. Consistently, the PISO (Fig. 7(b)) and SIMPLEST/PRIME (Fig. 7(f)) algorithms need the lowest and highest number of iterations, respectively. As in the previous two cases, the convergence attributes of SIMPLE (Fig. 7(c)), SIMPLEC (Fig. 7(d)), and SIMPLEX (Fig. 7(g)) are very similar and those of SIMPLEM (Fig. 7(e)) are close to them. The large difference in performance between SIMPLEST/PRIME and the remaining algorithms is clearly demonstrated in Fig. 7(h). The normalized CPU time of SIMPLEST/PRIME for this problem (Fig. 8) is lower than in the previous two problems due to a higher rate of increase in the time needed by other algorithms (the real computational time of all algorithms has increased). The relative performance of the various algorithms is nearly as described earlier with the time required by of PISO, SIMPLE, SIMPLEC, and SIMPLEX being on average the same. The SIMPLEST/PRIME algorithm however, requires nearly three folds the time needed by SIMPLE, which represents a noticeable improvement.

Dense bubbly flow

The only difference between this case and the previous one is in the concentration of bubbles, which is set to $r_{\text{inlet}}^{(\text{d})} = 0.5$. With such high value of void fraction, bubble coalescence may occur. However, this is not accounted for here. The analytical solution is the same as in the previous case with the equilibrium velocity, as computed from Eq. (108), being 3 m/s. As depicted in Fig. 9(a), the equilibrium velocity obtained numerically is exact. Moreover, the performance of the various algorithms (Figs. 9(b)-9(h)) and their normalized CPU times (Fig. 10) vary relatively in a manner similar to what was previously discussed and it is deemed redundant to be repeated.

Vertical particle transport

Here, the flow is in the vertical direction (Fig. 1) and the gravitational acceleration is assigned the constant value of $g=10 \text{ m/s}^2$ and the flow field is extended a length of $L=20\text{m}$. For this situation, the velocities of the two phases do not equilibrate to a constant. Rather, the disperse phase equilibrates to a finite settling velocity relative to the continuous phase, at which the gravitational force balances the drag force [81]. As for the horizontal transport problems, the inter-particle/bubble forces are neglected. However, unlike the previous situation, diffusion in the continuous phase is retained. Moreover, the inter-phase drag force is calculated using Eqs. (105)-(106) and the drag coefficient, C_D , is considered to be particle Reynolds number dependent and calculated as:

$$C_D = \frac{24}{Re_p} + 0.44 \quad (110)$$

where

$$Re_p = \frac{2r_p V_{\text{slip}}}{\vartheta^{(\text{c})}} \quad (111)$$

Since diffusion in the continuous phase is not neglected, the performances of MCBA-SIMPLEST and MCBA-PRIME are expected to be different.

Dilute gas-solid flow

The material properties and boundary conditions considered for this case are given by:

$$\rho^{(\text{d})}/\rho^{(\text{c})} = 1000, \vartheta^{(\text{c})} = 10^{-5}, r_p = 1 \text{ mm} \quad (112)$$

$$V_{\text{inlet}}^{(\text{c})} = 100 \text{ m/s}, V_{\text{inlet}}^{(\text{d})} = 10 \text{ m/s}, r_{\text{inlet}}^{(\text{d})} = 10^{-6} \quad (113)$$

The large velocity boundary conditions are used to ensure that the solid phase does not exit the inlet. The predicted air and particle velocity distributions depicted in Fig. 11(a) are in excellent agreement with similar predictions reported in [81]. As shown in Figs. 11(b)-11(h), the decrease in the mass residuals is highly non-monotonic showing a cyclic decaying behavior. The PISO algorithm (Fig. 11(b)) seems to be the least affected and its convergence path shows little cycling as compared to other algorithms and requiring the least number of iterations over all grids. It should be mentioned here that decreasing the under-relaxation factors could have smoothed this cycling behavior out. This, however, would have been accomplished at the expense of increasing the computational time. Even though the convergence is not monotonic, the number of iterations needed to converge the solution to the desired level is very close to that needed in the similar horizontal transport case. The convergence of the SIMPLEC algorithm over the dense grid system (Fig. 2(d)) shows cycles of large amplitudes due to the fact that pressure is not under-relaxed. The performance of SIMPLEST (Fig. 2(f)) and PRIME (Fig. 2(g)) is very close, with SIMPLEST requiring slightly lower number of iterations due to the small implicitness introduced by the diffusion of the continuous phase. However, both require on the finest mesh almost 1300% the number of iterations needed by SIMPLE. The number of outer iterations needed by SIMPLEX, SIMPLEC, and SIMPLEM is very close to that of SIMPLE, with SIMPLEM requiring the highest number of iterations. This behavior is further revealed by the normalized CPU times needed by the various algorithms that are displayed in Fig. 12. The efficiency of SIMPLEST is slightly better than PRIME, both however are about five times more expensive than all other algorithms whose performance is very comparable.

Dense gas-solid flows

The material properties and boundary conditions are similar to the previous case with the exception of the particles' volume fraction, which is set to $r_{\text{inlet}}^{(d)} = 10^{-2}$. Predicted air and particle velocity profiles are displayed in Fig. 13(a) while mass residuals are presented in Figs. 13(b)-13(h). The convergence paths are smoother in comparison with the dilute case due to the higher volume fraction of the disperse phase, which makes the computations less sensitive to the intermediate level of convergence. However, higher number of iterations is needed in comparison with the dilute case due to the higher

mass-loading ratio. Besides that, the convergence behavior is similar to the previous cases with SIMPLEST (Fig. 13(f)) and PRIME (Fig. 13(g)) requiring the highest number of iterations (SIMPLEST needs slightly less iterations for the reason stated above). The number of iterations needed by PISO, SIMPLEC, and SIMPLEX (compare Figs. 13(b), 13(d), and 13(h)) is very close. The SIMPLEM algorithm (Fig. 13(e)) takes a slightly higher number of iterations than SIMPLE (Fig. 13(c)). For this problem SIMPLEX requires the least computational time followed by SIMPLEC and SIMPLE (Fig. 14). Again the computational effort needed by SIMPLEST and PRIME is much higher (about 700% the time needed by SIMPLEX on the finest grid).

Dilute bubbly flows

In this problem, the continuous phase is water and the disperse phase is air. With the exception of $\rho^{(d)} / \rho^{(c)}$ set to 10^{-3} and $r_{inlet}^{(d)}$ to 0.1 at inlet, other physical properties and inlet conditions are the same as those considered earlier. This is a very difficult problem to get convergence to unless the proper under-relaxation is used. It was possible to get feasible solutions (i.e. with reasonable computational time) when under-relaxing by inertia (i.e. through the use of false time steps). For the results presented in Figs. 15 and 16, a time step (Δt) of value 10^{-4} s is used for the velocity field of the dispersed gas phase, $\Delta t=1$ s for the volume fractions, and $\Delta t=0.01$ s for the velocity field of the liquid phase (this last value is employed with all algorithms except PISO and PRIME for which a value of 0.1 s is used). Also, to accelerate convergence, it is found advantageous not to under-relax the pressure field. As expected, the correct velocity fields are predicted (Fig. 15(a)). All algorithms require higher number of iterations as compared to the previous problems. Moreover, convergence flattens after a certain level because of the large under-relaxation used for the bubble phase momentum equations. Using an automatic time stepping algorithm that increases this time step as the residuals are decreased could resolve this issue. The performance of PISO (Fig. 15(b)) is totally unexpected requiring higher number of iterations than SIMPLE (Fig. 15(c)), SIMPLEC (Fig. 15(d)), SIMPLEM (Fig. 15(e)), and SIMPLEX (Fig. 15(h)). The decrease in the rate of convergence as the residuals are decreased, is very clear with SIMPLE, SIMPLEC, and SIMPLEX which show a sudden change in their convergence slopes. The convergence histories of these algorithms are nearly

identical. The explicitness introduced in PISO (Fig. 15(b)), SIMPLEST (Fig. 15(f)), and PRIME (Fig. 15(g)) causes an increase in the number of iterations and a highly non-monotonic convergence behavior and seems to be undesirable in laminar bubbly flows. The rate of convergence of the SIMPLEM algorithm (Fig. 15(e)) is nearly constant and does not show any change in slope as the computations progress. This is due to the larger number of inner iterations needed on the pressure correction equation to promote convergence and is associated with higher computational cost as depicted in Fig. 16. The noticeable change in the normalized time chart is the performance of SIMPLEST which shows a good improvement and that of PISO, which has deteriorated. The CPU times needed by SIMPLE, SIMPLEC, and SIMPLEX are consistently very close. For a given grid however, the largest CPU time consumed is only double the lowest CPU needed.

Dense bubbly flows

With the exception of setting $r_{inlet}^{(d)}$ to 0.5, the physical situation, material properties, and boundary conditions are the same as in the previous problem. Results for the problem are presented in Figs. 17 and 18. The predicted liquid and gas velocity distributions, which are in excellent accord with published data, are depicted in Fig. 17(a). The trend of convergence (Figs. 17(b)-17(h)) is very similar to the dilute case with the following differences. With PISO and PRIME, in order to drive residuals to the desired level of convergence, the SMART scheme had to be blended with the UPWIND scheme. The percentage increased from 15% on the coarsest grid to 50% on the finest grid. Moreover, mass residuals reached the desired convergence level long before the momentum residuals. Thus, the numbers of iterations needed by PISO and PRIME are much higher than the ones presented in Figs. 17(b) and 17(g) and are reflected by the CPU times displayed in Fig. 18. In addition, to stabilize SIMPLEST on the finest grid, the SMART scheme had to be blended with 20% of the upwind value. The difficulties faced by PISO, PRIME, and SIMPLEST are attributed to the additional explicitness introduced as a result of using a HR scheme implemented via a deferred correction strategy [82]. This deferred correction technique has lower influence on the performance of the remaining algorithms due to their higher implicitness. As shown in Fig. 18, even with the blending strategy, PISO and PRIME still need a high computational time to decrease the residuals to the desired level. The SIMPLEST

algorithm requires the lowest computational effort among the three algorithms on the finest grid when blended with the upwind scheme. The CPU times consumed by SIMPLE, SIMPLEC, and SIMPLEX are consistently very close to each other.

By comparing the behavior of the various algorithms in all problems, it is clear that the performance of SIMPLE, SIMPLEC, and SIMPLEX is consistent and require, on average, the least computational effort. Even though in some cases PISO consumes less CPU time, its performance for upward bubbly flows is not satisfactory. The performance of SIMPLEM is consistent, but demands more CPU time than SIMPLE, SIMPLEC, and SIMPLEX. The SIMPLEST algorithm is recommended for diffusion-dominated flows (e.g. turbulent flows). Nevertheless, other algorithms may perform, in such situations, equally well if not better. The PRIME algorithm is the most expensive to use on all grids and for all physical situations presented here. Most importantly however, is the fact that all these algorithms can be used to predict multi-phase (in this case two-phase) flows. The next step is to test these algorithms in multi-dimensional incompressible, compressible, laminar, and turbulent flows.

Two-dimensional two-phase validation problems

In this section, four two-dimensional two-phase problems are solved. The first two problems deal with incompressible turbulent flows while the last two problems are concerned with compressible flows. As mentioned earlier, solutions will be obtained using the prolongation grid scheme (PG) and the full multi-grid method (FMG) in addition to the usual single grid (SG) approach. This will enable assessing the performance of the non-linear multi-grid with multi-phase flows.

Turbulent upward bubbly flow in a pipe

The problem considered involves the prediction of radial phase distribution in turbulent upward air-water flow in a pipe. Many experimental and numerical studies addressing this problem have appeared in the literature [83-91]. Most of these studies have indicated that the lateral forces that most strongly affect the void distribution are the turbulent stresses and the lateral lift force. As such, in addition to the usual drag force, the lift force is considered as part of the interfacial force terms in the momentum equations. In the present work, the interfacial drag forces per unit volume are given by:

$$(I_M^x)_D^{(c)} = -(I_M^x)_D^{(d)} = 0.375 \frac{C_D}{r_p} \rho^{(c)} r^{(d)} r^{(c)} V_{\text{slip}} (u^{(d)} - u^{(c)}) \quad (114)$$

$$(I_M^y)_D^{(c)} = -(I_M^y)_D^{(d)} = 0.375 \frac{C_D}{r_p} \rho^{(c)} r^{(d)} r^{(c)} V_{\text{slip}} (v^{(d)} - v^{(c)}) \quad (115)$$

where r_p is the bubble radius. The drag coefficient C_D varies as a function of the bubble Reynolds and Weber numbers defined as:

$$\begin{cases} Re = 2 \frac{r_p}{V_1^{(c)}} V_{\text{slip}} \\ We = 4 \rho^{(c)} \frac{r_p^2}{\sigma} V_{\text{slip}} \end{cases} \quad (116)$$

where σ is the surface tension, which is given a value of 0.072N/m for air-water systems. The following correlations, which take the shape of the bubble into consideration, are utilized [92,93]:

$$\begin{cases} C_D = \frac{16}{Re_p} & \text{for } Re_p < 0.49 \\ C_D = \frac{20}{Re_p^{0.643}} & \text{for } 0.49 < Re_p < 100 \\ C_D = \frac{6.3}{Re_p^{0.385}} & \text{for } Re_p >> 100 \\ C_D = \frac{8}{3} & \text{for } Re_p >> 100 \text{ and } We > 8 \\ C_D = \frac{We}{3} & \text{for } Re_p >> 100 \text{ and } Re_p > 2065.1/We^{2.6} \end{cases} \quad (117)$$

Many investigators have considered the modeling of lift forces [92-95]. Based on their work, the following expressions are employed for the calculation of the interfacial lift forces per unit volume:

$$(I_M)_L^{(c)} = -(I_M)_L^{(d)} = C_1 \rho^{(c)} r^{(d)} (u^{(d)} - u^{(c)}) \times (\nabla_x u^{(c)}) \quad (118)$$

where C_1 is the interfacial lift coefficient calculated from:

$$C_1 = C_{1a} (1 - 2.78 \langle 0.2, r^{(d)} \rangle) \quad (119)$$

where $\langle a, b \rangle$ denotes the minimum of a and b and C_{1a} is an empirical constant.

Besides the drag and lift interfacial forces, the effect of bubbles on the turbulent field is very important, since the distribution of bubbles affects the turbulence field in the liquid phase and at the same time the liquid phase's turbulence is influenced by the bubbles. In this work, turbulence is assumed to be a property of the continuous liquid phase (c) and the turbulent kinematic viscosity of

the dispersed air phase (d) is assumed to be a function of that of the continuous phase. The turbulent viscosity of the continuous phase is computed by solving the following modified transport equations for the turbulent kinetic energy k and its dissipation rate ε that take into account the interaction between the phases:

$$\frac{\partial(r^{(c)}\rho^{(c)}k^{(c)})}{\partial t} + \nabla \cdot (r^{(c)}\rho^{(c)}\mathbf{u}^{(c)}k^{(c)}) = \nabla \cdot \left[r^{(c)}\rho^{(c)} \left(v_l^{(c)} + \frac{v_t^{(c)}}{\sigma_k^{(c)}} \right) \nabla k^{(c)} \right] + r^{(c)}\rho^{(c)}(G^{(c)} - \varepsilon^{(c)}) + \nabla \cdot \left[\rho^{(c)} \left(\frac{v_t^{(c)}}{\sigma_r} \right) k^{(c)} \nabla r^{(c)} \right] + r^{(c)}P_b \quad (120)$$

$$\frac{\partial(r^{(c)}\rho^{(c)}\varepsilon^{(c)})}{\partial t} + \nabla \cdot (r^{(c)}\rho^{(c)}\mathbf{u}^{(c)}\varepsilon^{(c)}) = \nabla \cdot \left[r^{(c)}\rho^{(c)} \left(v_l^{(c)} + \frac{v_t^{(c)}}{\sigma_\varepsilon^{(c)}} \right) \nabla \varepsilon^{(c)} \right] + r^{(c)}c_{le}P_b \frac{\varepsilon^{(c)}}{k^{(c)}} + r^{(c)}\rho^{(c)} \frac{\varepsilon^{(c)}}{k^{(c)}} (c_{1\varepsilon}G^{(c)} - c_{2\varepsilon}\varepsilon^{(c)}) + \nabla \cdot \left[\rho^{(c)} \left(\frac{v_t^{(c)}}{\sigma_r} \right) \varepsilon^{(c)} \nabla r^{(c)} \right] \quad (121)$$

where $G^{(c)}$ is the well known volumetric production rate of $k^{(c)}$ by shear forces, σ_r the turbulent Schmidt number for volume fractions, and P_b is the production rate of $k^{(c)}$ by drag due to the motion of the bubbles through the liquid and is given by:

$$P_b = \frac{0.375C_b C_D \rho^{(c)} r^{(d)} r^{(c)} V_{slip}^2}{r_p} \quad (122)$$

In Eq. (122) C_b is an empirical constant representing the fraction of turbulence induced by bubbles that goes into large-scale turbulence of the liquid phase. Moreover, as suggested in [91], the flux representing the interaction between the fluctuating velocity and volume fraction is modeled via a gradient diffusion approximation and added as a source term in the continuity ($\nabla \cdot (\rho^{(k)} D^{(k)} \nabla r^{(k)})$) and momentum ($\nabla \cdot (\rho^{(k)} D^{(k)} \mathbf{u}^{(k)} \nabla r^{(k)})$) equations with the diffusion coefficient D given by:

$$D^{(k)} = \frac{v_t^{(k)}}{\sigma_r} \quad (123)$$

The turbulent kinematic eddy viscosity of the dispersed and continuous phases are related through:

$$v_t^{(d)} = \frac{v_t^{(c)}}{\sigma_f} \quad (124)$$

where σ_f is the turbulent Schmidt number for the interaction between the two phases. The above described turbulence model is not exactly the one described in [91] but rather a modified version of

that model in which the turbulent kinematic viscosities of both phases are allowed to be different in contrast to what is done in [91]. This is accomplished through the introduction of the σ_f parameter. As such, different diffusion coefficients ($D^{(k)}$) are used for the different phases.

To check the validity and correctness of the above described treatment two different situations, for which data are available, are considered. In the first, the data reported by Seriwaza et al [83] is reproduced. The second situation however, is concerned with the data of Lahey et al [94]. For the Seriwaza et al data [83], the Reynolds number based on superficial liquid velocity and pipe diameter is 8×10^4 , the inlet superficial gas and liquid velocities are 0.077 and 1.36 m/s, respectively, and the inlet void fraction is 5.36×10^{-2} with no slip between the incoming phases. Moreover, the bubble diameter is taken as 3 mm [91], while the fluid properties are taken as $\rho^{(c)}=1000$ Kg/m³, $\rho^{(d)}=1.23$ Kg/m³, and $v_l^{(c)}=10^{-6}$ m²/s. On the other hand, Lahey et al's data [94] is for a Reynolds number based on superficial liquid velocity and pipe diameter of 5×10^4 , the inlet superficial gas and liquid velocities are 0.1 and 1.08 m/s, respectively. Both problems are solved using the same values for all constants in the model with $C_{la}=0.075$, $\sigma_f=0.5$, $\sigma_r=0.7$, and $C_b=0.05$. Predicted radial profiles of the vertical liquid velocity and void fraction presented in Figs. 19(a) and 19(b) using a grid of size 96x32 control volumes concur very well with measurements and compare favorably with numerical profiles reported by Boisson and Malin [91] (Fig. 19(a)) and PHOENICS [96] (Fig. 19(b)).

Having established the credibility of the physical model and numerical procedure, the next task is to assess the merits of the various algorithms for such flows. For that purpose the Seriwaza et al [83] configuration is chosen and the axi-symmetric two-dimensional calculations are performed on 4 different grids of sizes 12x4, 24x8, 48x16, and 96x32 control volumes. On each grid, solutions are generated using the SG, PG, and FMG strategies for all algorithms. Results are displayed in the form of (i) total mass residuals summed over both phases as a function of outer iterations, and (ii) normalized CPU time (Table 1) needed for the maximum normalized residual of all variables and for all phases to drop below $\epsilon_s=10^{-6}$. In order not to overload plots, only residuals over the densest grid using the SG, PG, and FMG methodologies are presented in Fig. 20. Moreover, it should be clarified that in the case of PISO and PRIME the number of iterations needed to satisfy the convergence

criteria is much higher than the one presented (this will be reflected by their respective normalized CPU times displayed in Table 1) due to the slower convergence rate of the axial liquid momentum and turbulence kinetic energy equations, i.e. the convergence rate of these equations is much slower than the overall mass conservation equation. Since the overall mass residuals are presented, only the iterations needed to drive these residuals to the required level (10^{-6}) are displayed, even though at the state of convergence these residuals are much lower (of the order of 10^{-9}).

As can be seen from Fig. 20, it is possible to converge the solution to the desired level with all algorithms. Moreover, the performance of SIMPLE (Fig. 20(b)), SIMPLEC (Fig. 20(c)), SIMPLEST (Fig. 20(e)), and SIMPLEX (Fig. 20(g)) is very close. This is also demonstrated in Fig. 20(h) where the convergence histories using the FMG method are plotted for all algorithms. As shown, the plots for these algorithms are hard to distinguish. For all algorithms, the SG requires the largest number of iterations and the FMG the lowest number of iterations. The FMG method greatly reduces the number of iterations whereas reduction with the PG method is not significant. The normalized CPU times displayed in Table 1 further reveals this. As shown in the plots, the rate of convergence of the SG and PG methods diminishes as the iterations proceed. This is in contrast with the convergence behavior of the FMG method, which seems to maintain, more or less, the same convergence rate (on the finest level) as the iterations progress.

In Table 1, the normalized CPU-times required by the different algorithms to solve the problem on the various grids and with the various methodologies are presented. In addition, the ratio of the time needed by the SG method to the one needed by either the PG or the FMG method is displayed. This allows for a direct quantitative assessment of their acceleration rate. As depicted, the CPU effort increases for all methods and algorithms with increasing the grid size. The CPU times of SIMPLE, SIMPLEC, SIMPLEX, and SIMPLEST are very on all grids and for all methods with no clear superiority to any algorithm over the others. PISO requires the least computational time on the coarsest grid used (12x4 C.V.) however, as the grid size increases the picture is reversed and its performance degrades and become closer to that of PRIME and SIMPLEM. The use of the PG method on the relatively coarse grids increases the computational cost. The benefit of the PG method is realized on the relatively dense grids. On the densest grid, the savings in computational effort vary

from 11% for PRIME to 28% for SIMPLEM with an average acceleration rate for all algorithms of 22%. The use of the FMG method is highly beneficial and its virtues generally appear on all grids with the rate of acceleration increasing with increasing grid density. On the finest grid, the rate of acceleration varies from 144% for SIMPLEM to 555% with PRIME with an average acceleration rate for all algorithms of 444%. This average rate of acceleration is nearly 20 times the one obtained with the PG method. This indicates that the significant non-linearity in multi-fluid flows is properly resolved using the non-linear multi grid method. Moreover, turbulence could be the reason behind the poor performance of the PG method.

Turbulent air-particle flow in a vertical pipe

The problem consists of predicting the upward flow of a dilute gas-solid mixture in a vertical pipe using the various algorithms and calculation procedures. For these simulations, the axi-symmetric form of the gas and particulate transport equations are employed. As reported in several studies [97-99], the effects of interfacial virtual mass and lift forces are small and may be neglected. The controlling interfacial force is drag and the closure used to compute it in the present study is due to Harlow and Amsden [100] in which the viscous part is neglected. Denoting the continuous and dispersed phases by superscripts (c) and (d), respectively, the drag in the x- and y-momentum equations are given by:

$$\left(I_M^x \right)_D^{(c)} = -\left(I_M^x \right)_D^{(d)} = \frac{3}{8} \frac{C_D}{r_p} \rho^{(c)} r^{(d)} V_{\text{slip}} (u^{(d)} - u^{(c)}) \quad (125)$$

$$\left(I_M^y \right)_D^{(c)} = -\left(I_M^y \right)_D^{(d)} = \frac{3}{8} \frac{C_D}{r_p} \rho^{(c)} r^{(d)} V_{\text{slip}} (v^{(d)} - v^{(c)}) \quad (126)$$

where r_p represents the particle's radius, C_D the drag coefficient computed from:

$$\begin{cases} C_D = \frac{24}{Re_p} & \text{for } Re_p < 1 \\ C_D = \frac{24}{Re_p} (1 + 0.15 Re_p^{0.687}) & \text{for } 1 < Re_p < 1000 \\ C_D = 0.44 & \text{for } Re_p > 1000 \end{cases} \quad (127)$$

and Re_p the Reynolds number based on the particle size as defined in Eq.(111).

As before, turbulence is assumed to be a property of the continuous gas phase (c) and the turbulent kinematic viscosity of the dispersed particle phase (d) is assumed to be a function of that of the continuous phase. Turbulence modulation due to the presence of particles is predicted using a two-phase k- ε model. Several extensions of the k- ε model for carrier-phase turbulence modulation have been proposed in the literature [70-76] and the one suggested by Chen and Wood [72], which introduces additional source terms into the turbulence transport equations, is adopted here. These source terms are always negative and act to reduce k and ε . However, depending on the relative extent of reductions in k and ε , the turbulent viscosity may be either reduced or increased by the presence of particles. Thus, the turbulent viscosity is computed by solving the turbulence transport equations (Eqs. (5) and (6)) for the continuous phase with $I_k^{(k)}$ and $I_\varepsilon^{(k)}$ evaluated using the following relations suggested by Chen and Wood [72]:

$$I_k^{(c)} = -2\rho^{(d)} r^{(c)} r^{(d)} \frac{k^{(c)}}{\tau_p} \left(1 - e^{-0.0825 \frac{\tau_p}{\tau_e}} \right) \quad (128)$$

$$I_\varepsilon^{(c)} = -2\rho^{(d)} r^{(c)} r^{(d)} \frac{\varepsilon^{(c)}}{\tau_p} \quad (129)$$

where τ_p and τ_e are timescales characterizing the particle response and large-scale turbulent motion, respectively, and are computed from:

$$\begin{cases} \tau_p = \frac{\rho^{(d)} r^{(d)}}{F_D} V_{\text{slip}} \\ \tau_e = 0.165 \frac{k^{(c)}}{\varepsilon^{(c)}} \end{cases} \quad (130)$$

where F_D is the magnitude of the inter-phase drag force per unit volume. The turbulent kinematic eddy viscosity of the dispersed phase is found using Eq. (124).

The model is validated against the experimental results of Tsuji et al [97]. In their experiments, the vertical pipe has an internal diameter of 30.5 mm. Results are replicated here for the case of an air Reynolds number, based on the pipe diameter, of 3.3×10^4 and a mean air inlet velocity of 15.6 m/s using particles of diameter 200 μm and density 1020 Kg/m^3 . In the computations, the mass-loading ratio at inlet is considered to be 1 with no slip between the phases, and σ_f and σ_r are set to 5 and 10^{10} (i.e. the interaction terms included for bubbly flows are neglected here), respectively. Figure 21 shows

the fully developed gas and particles mean axial velocity profiles generated using a grid of size 96x40 C.V. It is evident that there is generally a very good agreement between the predicted and experimental data with the gas velocity being slightly over predicted and the particles velocity slightly under predicted. Moreover, close to the wall, the model predictions indicate that the particles have higher velocities than the gas, which is in accord with the experimental results of Tsuji et al. [97].

After establishing the credibility of the physical model and numerical procedure, the problem is solved over four different grids of sizes 12x5, 24,x10, 48x20, and 96x40 control volumes using the SIMPLE, SIMPLEC, SIMPLEX, and SIMPLEST multi-fluid algorithms and the SG, PG, and FMG solution methods. In solving he problem, it is noticed that the initial guess greatly affects the convergence history and time required to converge the solution to the desired level. Except when solving on the finest mesh with the SG method, the initial guess used for the velocity field is $u^{(c)}=u^{(d)}=1$ m/s. The use of this initial value with the SG method on the finest mesh increased the CPU effort needed by 500% over the one needed when starting with an initial field of $u^{(c)}=u^{(d)}=15.6$ m/s. To reduce cost, the latter initial guess is used. For this reason the mass residuals start from somehow a lower value than expected. As in the previous problem, results are displayed in the form of (i) total mass residuals summed over both phases as a function of outer iterations (Fig. 22), and (ii) normalized CPU time (Table 2) needed for the maximum normalized residual of all variables and for all phases to drop below $\epsilon_s=10^{-6}$. In order not to overload plots, only residuals over the densest grid using the SG, PG, and FMG methodologies are presented. As shown (Fig. 22), the PG and FMG methods greatly reduce the number of iterations with the number required by the PG method being around four times the number required by he FMG method. The slopes of the various curves displayed in Fig. 22, which are clearly much higher for the FMG method, further reveal this.

The normalized CPU-times, required by the four algorithms to solve the problem on the various grids and with the various methodologies, presented in Table 2 indicate an increase with increasing the grid size. Moreover, higher savings are realized with both the PG and FMG methods over the ones obtained in the previous problem over all grids. Again the CPU-time required by the four algorithms is very close with no appreciable superiority of any one. Using the PG method on the finest mesh

accelerates the convergence by a factor of 3.6 whereas the FMG does a much better job of accelerating the solution by a factor of around 9.

Compressible dilute air-particle subsonic flow over a flat plate

This problem has been used by several researchers [101-107] to check their computational methodologies and is used here for the same purpose. It is known that two-phase flow greatly changes the main features of the boundary layer over a flat plate. Typically, three different regions are defined in the two-phase boundary layer (Fig. 23), which can be distinguished by the relative velocity between the two phases: a large-slip region close to the leading edge, a moderate-slip region further down, and a small-slip one far downstream. It has also been established that [104], the characteristic scale in this two-phase problem is the relaxation length λ_e , defined as:

$$\lambda_e = \frac{2}{9} \frac{\rho^{(d)} r_p^2 u_\infty}{\mu^{(c)}} \quad (131)$$

where $\rho^{(d)}$ and r_p are respectively the density and radius of the particles, $\mu^{(c)}$ the viscosity of the fluid, and u_∞ the free stream velocity. The three regions are defined according to the order of magnitude of the slip parameter $x^* = x / \lambda_e$. In the simulation, the viscosity of the fluid is considered to be a function of temperature and varies according to [104]:

$$\mu^{(c)} = \mu_{ref} \left(\frac{T^{(c)}}{T_{ref}} \right)^{0.6} \quad (132)$$

where the reference viscosity and temperature are $\mu_{ref} = 1.86 \times 10^{-5}$ N.s/m² and $T_{ref} = 303$ °K, respectively. Even though variations in gas density are small under the conditions considered, these variations are not neglected and the flow is treated as compressible for the continuous phase and as incompressible for the dispersed phase. Moreover, drag is the only interfacial force retained due to its dominance over other interfacial forces. Denoting the continuous and dispersed phases by superscripts (c) and (d), respectively, this force is computed as [104]:

$$(I_M^x)_D^{(c)} = -(I_M^x)_D^{(d)} = \frac{9}{2} \frac{C_D}{r_p^2} r^{(d)} \mu^{(c)} (u^{(d)} - u^{(c)}) \quad (133)$$

$$(I_M^y)_D^{(c)} = -(I_M^y)_D^{(d)} = \frac{9}{2} \frac{C_D}{r_p^2} r^{(d)} \mu^{(c)} (v^{(d)} - v^{(c)}) \quad (134)$$

where the drag coefficient is given by:

$$C_D = \frac{1}{50} Re_p + \frac{7}{6} Re_p^{0.15} \quad (135)$$

In the energy equation, heat transfer due to radiation is neglected and only convective heat transfer around an isolated particle is considered. Moreover, the particles have no individual random motion, mutual collisions, and other interactions among them. Therefore, only the process of drag and heat transfer couple the particles with the gas. Under such conditions, the interfacial terms in the gas (continuous phase) and particles (dispersed phase) energy equations reduce to [104]:

$$I_E^{(c)} = Q_{g-p} + \mathbf{F}_{g-p} \cdot \mathbf{u}^{(d)} \quad (136)$$

$$I_E^{(d)} = -Q_{g-p} \quad (137)$$

where:

$$\mathbf{F}_{g-p} = (I_M^x)_D^{(c)} \mathbf{i} + (I_M^y)_D^{(c)} \mathbf{j} \quad (138)$$

$$Nu = 2.0 + 0.6 Re_p^{\frac{1}{2}} (Pr^{(c)})^{\frac{1}{3}} \quad (139)$$

$$Q_{g-p} = \frac{3}{2} \frac{r^{(d)} \lambda^{(c)} Nu}{r_p^2} (T^{(d)} - T^{(c)}) \quad (140)$$

In the above equations, Nu is the Nusselt number, $Pr^{(c)}$ the gas Prandtl number, $\lambda^{(c)}$ the gas thermal conductivity, T the temperature, and other parameters are as defined earlier.

In the simulation, the particle diameter is chosen to be 10 μm , the particle Reynolds number is assumed to be equal to 10, and the material density is 1766 kg/m^3 . The Prandtl number is equal to 0.75. The south boundary (wall) is treated as a no-slip wall boundary for the gas phase: both components of the gas velocity are set to zero, while the particle phase encounters slip wall conditions. The normal fluxes are set to zero. The gas and the particles enter the computational domain under thermal and dynamical equilibrium conditions. A mass load ratio of 1 between the particles phase and the gas phase is used. Results are displayed using the following dimensionless variables in order to bring all quantities to the same order of magnitudes:

$$x^* = \frac{x}{\lambda_e}, y^* = \frac{y}{\lambda_e} \sqrt{Re}, u^* = \frac{u}{u_\infty}, v^* = \frac{v}{u_\infty} \sqrt{Re} \quad Re = \frac{\rho u \lambda_e}{\mu} \quad (141)$$

Figure 24 shows the results for the steady flow obtained on a rectangular domain with a mesh of density 104x48 C.V. stretched in the y-direction. The figure provides the development of gas and

particles velocity profiles within the three regions mentioned earlier. In the near leading edge area ($x^*=0.1$), the gas velocity is adjusted at the wall to obtain the no-slip condition as for the case of a pure gas boundary layer. The particles have no time to adjust to the local gas motion and there is a large velocity slip between the phases. In the transition region ($x^*=1$), significant changes in the flow properties take place. The interaction between the phases cause the particles to slow down while gas accelerate as apparent in the plots. In the far downstream region ($x^*=5$), the particles have enough time to adjust to the state of the gas motion. The slip is very small and the solution tends to equilibrium. These results are in excellent agreement with numerical solutions reported by Thevand et al. [107] plotted in Fig. (24), which gives credibility to the proposed methodology.

Credibility being established, the problem is solved over four different grids of sizes 13x6, 26,12, 52x24, and 104x48 control volumes using all multi-fluid algorithms and the SG, PG, and FMG solution methods. As in the previous two problems, results are displayed in the form of total mass residuals (Fig. 25) and normalized CPU time (Table 3) with $\epsilon_s=10^{-6}$. Only residuals over the densest grid using the SG, PG, and FMG methodologies are presented.

As depicted in Fig. 25(a), PISO requires the least number of iterations for the reasons stated earlier (the same apply for PRIME Fig. 25 (f)) and this is not associated with the lowest computational effort (Table 3). Moreover, the convergence behavior of SIMPLE (Fig. 25(b)), SIMPLEC (Fig. 25(c)), SIMPLEM (Fig. 25(d)), and SIMPLEX (Fig. 25(g)) is very similar requiring nearly the same number of iterations with all solution methods. In comparison, SIMPLEST (Fig. 25(e)) consumes a larger number of iterations. The worst performance is for PRIME (Fig. 25(f)) especially with the SG method where it is seen to require over 8000 iterations to reduce the total mass residuals to the desired level. The above can also be inferred from the plots displayed in Fig. 25(h). Again, the PG and FMG methods greatly reduce the number of iterations with the number of iterations required by the PG method, for most algorithms, being around four times the number required by the FMG method.

The normalized CPU-times for all cases considered are presented in Table 3. The general trend is similar to the previous cases with the SG requiring the highest CPU effort and the FMG being the most efficient. However, the PG method is not as efficient as in the previous problem whereas the FMG remains reliably competent. In fact, the use of the PG method on the dense grid decreases, on

average for all algorithms, the computational time by about 37% whereas about 570% reduction in computational time is realized with the FMG method. Consistently, the CPU-time required by the SIMPLE, SIMPLEC, and SIMPLEX algorithms is very close with no appreciable superiority of any one of these algorithms over the others. Moreover, the virtues of the FMG method increase as the grid size increases (e.g. for SIMPLEC the acceleration rate increases from 1.2 with two-levels, to 2.26 with three-levels, up to 8.06 with four levels). On the dense grids, which is of interest, the most time-consuming algorithm is PRIME followed by SIMPLEST and PISO. The best performance however, is for SIMPLE, SIMPLEC, and SIMPLEX with that of SIMPLEM being close.

Inviscid transonic dusty flow in a converging-diverging nozzle

As a final test for the newly suggested numerical procedures, dilute two-phase supersonic flow in an axi-symmetric converging-diverging rocket nozzle is considered. Several researchers have analyzed the problem and data is available for comparison [108-117]. In most of the reported studies, a shorter diverging section, in comparison with the one considered here, has been used when predicting the two-phase flow. Two-phase results for the long configuration have only been reported by Chang et al. [112]. The flow is assumed to be inviscid and the single-phase results are used as an initial guess for solving the two-phase problem. The physical configuration (Fig. 26) is the one described in [112]. The viscosity of the fluid varies with the temperature according to Sutherland's law for air:

$$\mu^{(c)} = 1.458 \times 10^{-6} \frac{T^{(c)} \sqrt{T^{(c)}}}{T^{(c)} + 110.4} \quad (142)$$

The coupling between gas and particle phases is through the interfacial momentum and energy terms.

The force exerted on a single particle moving through a gas is given as [113]

$$f_x = 6\pi r_p f_D \mu^{(c)} (u^{(d)} - u^{(c)}) \quad (143)$$

$$f_y = 6\pi r_p f_D \mu^{(c)} (v^{(d)} - v^{(c)}) \quad (144)$$

so that for N particles in a unit volume the effective drag force is

$$(I_M^x)_D^{(c)} = -(I_M^x)_D^{(d)} = \frac{9}{2} \frac{r^{(d)}}{r_p^2} f_D \mu^{(c)} (u^{(d)} - u^{(c)}) \quad (145)$$

$$(I_M^y)_D^{(c)} = -(I_M^y)_D^{(d)} = \frac{9}{2} \frac{r^{(d)}}{r_p^2} f_D \mu^{(c)} (v^{(d)} - v^{(c)}) \quad (146)$$

where f_D is the ratio of the drag coefficient C_D to the stokes drag $C_{D0}=24/\text{Re}_p$ and is given by [112]

$$f_D = 1 + 0.15 Re_p^{0.687} + \frac{0.0175 Re_p}{1 + 4.25 \times 10^4 Re_p^{-1.16}} \quad Re_p < 3 \times 10^5 \quad (147)$$

The heat transferred from gas to particle phase per unit volume is given as [113]

$$Q_{g-p} = \frac{3 r^{(d)}}{2 r_p} \lambda^{(c)} \text{Nu} (T^{(d)} - T^{(c)}) \quad (148)$$

Where $\lambda^{(c)}$ is the thermal conductivity of the gas and Nu, the Nusselt number, is written as [113]

$$\text{Nu} = 2 + 0.459 Re_p^{0.55} Pr_c^{0.33} \quad (149)$$

The gas-particle inter-phase energy term is given by

$$I_E^{(c)} = \frac{9}{2} \frac{r^{(d)}}{r_p^2} f_D \mu^{(c)} (u^{(d)} - u^{(c)}) u_d + \frac{9}{2} \frac{r^{(d)}}{r_p^2} f_D \mu^{(c)} (v^{(d)} - v^{(c)}) v_d + \frac{3}{2} \frac{r^{(d)}}{r_p} \lambda^{(c)} \text{Nu} (T^{(d)} - T^{(c)}) \quad (150)$$

$$I_E^{(d)} = \frac{3}{2} \frac{r^{(d)}}{r_p} \lambda^{(c)} \text{Nu} (T^{(c)} - T^{(d)}) \quad (151)$$

where the first two terms on the right-hand side of equation (150) represent the energy exchange due to momentum transfer.

The physical quantities employed are similar to those used in [112]. The gas stagnation temperature and pressure at inlet to the nozzle are 555 °K and 10.34×10^5 N/m², respectively. The specific heat for the gas and particles are 1.07×10^3 J/Kg°K and 1.38×10^3 J/Kg°K, respectively, and the particle density is 4004.62 kg/m³. With a zero inflow velocity angle, the fluid is accelerated from subsonic to supersonic speed in the nozzle. The inlet velocity and temperature of the particles are presumed to be the same as those of the gas phase. Results for two particle sizes of radii 1 and 10 μm with the same mass fraction $\phi=0.3$ are presented using a grid of size 188x80 C.V. Figures 27(a) and 27(b) show the particle volume fraction contours while Figures 27(c) and 27(d) display the velocity distribution. For the flow with particles of radius 1 μm, a sharp change in particle density is obtained near the upper wall downstream of the throat, and the particle density decreases to a small value. With the large particle flow (10 μm), however, a much larger particle-free zone appears due to the inability of the heavier particles to turn around the throat corner. These findings are in excellent agreement with published results reported by Chang et al. [112] and others using different methodologies. In addition contours are similar to those reported by Chang et al. [112]. A quantitative comparison of current predictions with published experimental and numerical data is presented in Fig. 28 through gas Mach

number distributions along the wall (Fig. 28(a)) and centerline (Fig. 28(b)) of the nozzle for the one-phase and two-phase flow situations with particles of radii 10 μm . As can be seen, the one-phase predictions fall on top of experimental data reported in [115-117]. Along the centerline of the nozzle, current predictions are of quality better than those obtained by Chang et al. [112]. Since the nozzle contour has a rapid contraction followed by a throat with a small radius of curvature, the flow near the throat wall is overturned and inclined to the downstream wall. A weak shock is thus formed to turn the flow parallel to the wall. This results in a sudden drop in the Mach number value and as depicted in Fig. 28(b), this sudden drop is correctly envisaged by the solution algorithm with the value after the shock being slightly over predicted.

Due to the unavailability of two-phase flow data, predictions are compared against the numerical results reported in [112]. As displayed in Figs. 28(a) and 28(b), both solutions are in good agreement with each other indicating once more the correctness of the calculation procedures. The lower gas Mach number in the two-phase flow is caused by the heavier particles ($\rho^{(d)} \gg \rho^{(c)}$), which reduce the gas velocity. Moreover, owing to the particle-free zone, the Mach number difference between the one- and two-phase flows along the wall is smaller than that at the centerline.

To compare the relative performance of the multi-fluid algorithms, the problem is solved via the SG method using the SIMPLE, SIMPLEC, SIMPLEX, and SIMPLEST algorithms over three different grids of sizes 47x20, 94x40, and 188x80 C.V. As before, results are displayed in the form of total mass residuals (Fig. 29) and normalized CPU times (Table 4) with $\epsilon_s = 10^{-6}$.

As shown in Fig. 29, with the exception of SIMPLEST on the 94x40 grid, all algorithms require almost the same number of iterations with SIMPLE and SIMPLEC being slightly better. As expected, the number of iterations increases with increasing grid density. Moreover, the convergence histories of all algorithms are nearly identical. The normalized CPU-times presented in Table 4 confirm these conclusions and reveal the close performance of SIMPLE and SIMPLEC. The normalized CPU-times needed by SIMPLEX are the highest due to the additional equations solved. Its performance could improve on denser grids.

Closing Remarks

The implementation of seven MCBA multi-fluid algorithms for the simulation of multi-fluid flow at all speeds was accomplished. The algorithms were embedded within a non-linear full multi-grid strategy. A two-fluid $k-\epsilon$ model and several inter-phase models were also employed. Solving a variety of one- and two-dimensional two-phase flow problems assessed the performance and accuracy of these algorithms. For each test problem, solutions were generated on a number of grid systems using the single grid method (SG), the prolongation grid method (PG), and the full non-linear multi-grid method (FMG). Results obtained demonstrated the capability of all algorithms to deal with multi-fluid flow situations and to predict multi-fluid flow at all speeds, and the ability of the FMG method to tackle the added non-linearity of laminar and turbulent multi-fluid flows. The convergence history plots and CPU-times presented, indicated similar performances for SIMPLE, SIMPLEC, and SIMPLEX. The PISO, SIMPLEM, and SIMPLEST algorithms were in general more expensive than SIMPLE. In general, the PRIME algorithm was the most expensive to use. The PG and FMG methods accelerated the convergence rate for all algorithms. The FMG method was found to be by far more efficient.

It is the author's view that the accomplishments are immensely greater than initially anticipated however there remains a great deal of work yet to be done. The items demanding additional efforts in the immediate future are:

1. Implementation of the GCBA group of algorithms
2. Implementation of more multi-fluid models (boiling, condensation, solidification, ...)
3. Implementation of additional turbulence models (e.g. $k-\omega$, SST,...)
4. Implementation of multi-component physics for the individual fluids
5. The use of unstructured grids

References

1. Zhu, J. and Leschziner, M.A., "A Local Oscillation-Damping Algorithm for Higher Order Convection Schemes," Computer Methods in Applied Mechanics and Engineering, vol. 67, pp. 355-366, 1988.
2. Gaskell, P.H. and Lau, A.K.C., "Curvature compensated Convective Transport: SMART, a new boundedness preserving transport algorithm," Int. J. Num. Meth. Fluids, vol. 8, pp. 617-641, 1988.
3. Darwish, M.S., "A new High-Resolution Scheme Based on the Normalized Variable Formulation," Numerical Heat Transfer, Part B: Fundamentals, vol. 24, pp. 353-371, 1993.
4. Chapman, M., "FRAM Nonlinear Damping Algorithm for the Continuity Equation," Journal of Computational Physics, vol. 44, pp. 84-103, 1981.
5. Zalesak, S.T., "Fully Multidimensional Flux-Corrected Transport Algorithm for Fluids," Journal of Computational Physics, vol. 31, pp. 335-362, 1979.
6. Harten, A., "High Resolution Schemes for Hyperbolic Conservation Laws," Journal of Computational Physics, vol. 49, no. 3, pp. 357-393, 1983.
7. Leonard, B.P., "Locally Modified Quick Scheme for Highly Convective 2-D and 3-D Flows," Taylor, C. and Morgan, K. (eds.), Numerical Methods in Laminar and Turbulent Flows, Pineridge Press, Swansea, U.K., vol. 15, pp. 35-47, 1987.
8. Darwish, M.s. and Moukalled, F., " Normalized Variable and Space Formulation Methodology For High-Resolution Schemes," Numerical Heat Transfer, Part B, vol. 26, pp. 79-96, 1994.
9. Moukalled, F. and Darwish ,M.S., "A New Bounded-Skew Central Difference Scheme- Part I: Formulation & Testing" Numerical Heat Transfer, Part B: Fundamentals, vol. 31, pp. 91-110, 1996.
10. Moukalled, F. and Darwish, M.S., "A New Bounded-Skew Central Difference Scheme- Part II: Application to Natural Convection in an Eccentric Horizontal Cylinder" Numerical Heat Transfer, Part B: Fundamentals, vol. 31, pp. 111-133, 1996.
11. Moukalled, F. and Darwish, M.S., "A New Family of Streamline-Based Very High Resolution Schemes" Numerical Heat Transfer, vol 32 No 3, pp. 299-320, 1997.
12. Darwish, M. and Moukalled, F., "An Efficient Very High-Resolution scheme Based on an Adaptive-Scheme Strategy," Numerical Heat Transfer, Part B, vol. 34, pp. 191-213, 1998.
13. Moukalled, F. and Darwish, M., "New Family of Adaptive Very High Resolution Schemes," Numerical Heat Transfer, Part B, vol. 34, pp. 215-239, 1998.
14. Patankar, S.V. and Spalding, D.B., "A Calculation Procedure for Heat Mass and Momentum Transfer in Three Dimensional Parabolic Flows", Int. J. Heat & Mass Trans., vol. 15, pp. 1787, 1972
15. Patankar, S.V.,Numerical Heat Transfer and Fluid Flow, Hemisphere, N.Y., 1981.
16. Issa, R.I., "Solution of the Implicit Discretized Fluid Flow Equations by Operator Splitting," Mechanical Engineering Report, FS/82/15, Imperial College, London, 1982.
17. Van Doormaal, J. P. and Raithby, G. D."Enhancement of the SIMPLE Method for Predicting Incompressible Fluid Flows' Numerical Heat Transfer, vol. 7, pp. 147-163, 1984.
18. Jang, D.S., Jetli, R. and Acharya, S., "Comparison of the PISO, SIMPLER and SIMPLEC Algorithms for the Treatment of the Pressure-Velocity Coupling in Steady Flow Problems," Numerical Heat Transfer, vol. 10, pp. 209-228, 1986.

19. Van Doormaal, J. P. and Raithby, G. D."An Evaluation of the Segregated Approach for Predicting Incompressible Fluid Flows," ASME Paper 85-HT-9, Presented at the National Heat Transfer Conference, Denver, Colorado, August 4-7, 1985.
20. Moukalled, F. and Darwish, M."A Unified Formulation of the Segregated Class of Algorithms for Fluid Flow at All Speeds," Numerical Heat Transfer, Part B, (in print).
21. Kershaw, D., "The Incomplete Cholesky-Conjugate Gradient Method for The Iterative Solution of Systems of Linear Equations," Journal of Computational Physics, vol. 26, pp. 43-65, 1978.
22. Stone, H.L., "Iterative Solution of Implicit Approximations of Multidimensional Partial Differential Equations," SIAM J. Numer. Anal., vol. 5, No. 3, pp. 530-558, 1968.
23. Brandt, A., "Multi-Level Adaptive Solutions to Boundary-Value Problems," Math. Comp., vol. 31, pp. 333-390, 1977.
24. Rhie, C.M., "A Pressure Based Navier-Stokes Solver Using the Multigrid Method," AIAA paper 86-0207, 1986.
25. Shyy, W. and Chen, M.H., "Pressure-Based Multigrid Algorithm for Flow at All Speeds," AIAA Journal, vol. 30, no. 11, pp. 2660-2669, 1992.
26. Shyy, W. and Braaten, M.E., "Adaptive Grid Computation for Inviscid Compressible Flows Using a Pressure Correction Method," AIAA Paper 88-3566-CP, 1988.
27. Rhie, C.M., "A Pressure Based Navier-Stokes Solver Using the Multigrid Method," AIAA paper 86-0207, 1986.
28. Yang, H.Q., Habchi, S.D., and Przekwas, A.J., "General Strong Conservation Formulation of Navier-Stokes Equations in Non-orthogonal Curvilinear Coordinates," AIAA Journal, vo. 32, no. 5, pp. 936-941, 1994.
29. Marchi, C.H. and Maliska, C.R., "A Non-orthogonal Finite-Volume Methods for the Solution of All Speed Flows Using Co-Located Variables," Numerical Heat Transfer, Part B, vol. 26, pp. 293-311, 1994.
30. Demirdzic, I., Lilek, Z., and Peric, M., "A Collocated Finite Volume Method For Predicting Flows at All Speeds," International Journal for Numerical Methods in Fluids, vol. 16, pp. 1029-1050, 1993.
31. Lien, F.S. and Leschziner, M.A., "A Pressure-Velocity Solution Strategy for Compressible Flow and Its Application to Shock/Boundary-Layer Interaction Using Second-Moment Turbulence Closure," Journal of Fluids Engineering, vol. 115, pp. 717-725, 1993.
32. Lien, F.S. and Leschziner, M.A., "A General Non-Orthogonal Collocated Finite Volume Algorithm for Turbulent Flow at All Speeds Incorporating Second-Moment Turbulence-Transport Closure, Part 1: Computational Implementation," Computer Methods in Applied Mechanics and Engineering, vol. 114, pp. 123-148, 1994.
33. Politis, E.S. and Giannakoglou, K.C., "A Pressure-Based Algorithm for High-Speed Turbomachinery Flows," International Journal for Numerical Methods in Fluids, vol. 25, pp. 63-80, 1997.
34. Chen, K.H. and Pletcher, R.H., "Primitive Variable, Strongly Implicit Calculation Procedure for Viscous Flows at All Speeds," AIAA Journal, vol.29, no. 8, pp. 1241-1249, 1991.
35. Issa, R.I. and Javarehkian, M.H., "Pressure-Based Compressible Calculation Method Utilizing Total Variation Diminishing Schemes," AIAA Journal, Vol. 36, No. 9, 1998.
36. Karimian, S.M.H. and Schneider, G.E., "Pressure-Based Control-Volume Finite Element Method for Flow at All Speeds," AIAA Journal, vol. 33, no. 9, pp. 1611-1618, 1995.
37. Schneider, G.E. and Kaimian, S.M.H., "Advances in Control-Volume-Based Finite-Element Methods for Compressible Flows," Computational Mechanics, Vol. 14, No. 5, pp. 431-446,

- 1994.
38. Karimian, S.M.H. and Schneider, G.E., "Pressure-Based Computational Method for Compressible and Incompressible Flows," AIAA Journal of Thermophysics and Heat Transfer, Vol. 8, No. 2, pp. 267-274, 1994.
 39. Darbandi, M. and Schneider, G.E., "Momentum Variable Procedure for Solving Compressible and Incompressible Flows," AIAA Journal, Vol. 35, No. 12, pp. 1801-1805, 1997.
 40. Darbandi, M. and Schneider, G.E., "Use of a Flow Analogy in Solving Compressible and Incompressible Flows," AIAA Paper 97-2359, Jan. 1997.
 41. Venkatakrishnan V., Mavriplis D.J. "Implicit Method for the Computation of Unsteady Flows on Unstructured Grids", AIAA paper 95-1705, 1995
 42. Venkatakrishnan V, "Perspective on Unstructured Grid Solvers", AIAA Journal, vol. 34, No. 3, pp. 533-547, 1996.
 43. Whitaker D.L., "Three Dimensional Unstructured Grid Euler Computations Using a Fully-Implicit, Upwind Method", AIAA paper 93-3357, 1993
 44. Oan D. Cheng J.C, "Upwind Finite Volume Navier-Stokes Computations on Unstructured Triangular Meshes", AIAA j., vol 31 No. 9, pp. 1618-1625, 1993
 45. Mitchell C.R. "Improved Reconstruction Schemes for the Navier-Stokes Equations on Unstructured Grids, AIAA paper 94-0642, 1994.
 46. Barth T.J, Jespersen D.C., "The Design and Application of Upwind Schemes on Unstructured Meshes", AIAA paper 89-0366, Jan. 1989
 47. Karki, K.C., "A Calculation Procedure for Viscous Flows at All Speeds in Complex Geometries," Ph.D. Thesis, University of Minnesota, June 1986.
 48. Moukalled, F. and Darwish, M., "A High-Resolution Pressure-Based Algorithm for Fluid Flow at All Speeds," Journal of Computational Physics (in print).
 49. Spalding D. B. 'Mathematical Modelling of Fluid Mechanics, Heat Transfer and Mass Transfer Processes,' Mech. Eng. Dept., Rept. HTS/80/1, Imperial College of Science, Technology and Medicine, London, 1980.
 50. Acharya, S. and Moukalled, F., "Improvements to Incompressible Flow Calculation on a Non-Staggered Curvilinear Grid," Numerical Heat Transfer, Part B, vol. 15, pp. 131-152, 1989.
 51. Maliska, C.R. and Raithby, G.D., "Calculating 3-D fluid Flows Using non-orthogonal Grid," Proc. Third Int. Conf. on Numerical Methods in Laminar and Turbulent Flows, Seattle, pp. 656-666, 1983.
 52. Spalding D.B., "The Calculation of Free-Convection Phenomena in Gas-Liquid Mixtures" Report HTS/76/11 Mech. Eng. Imperial College, London, 1976
 53. Spalding, D.B., "Numerical Computation of Multi-Phase Fluid Flow and Heat Transfer", in Recent Advances in Numerical Methods in Fluid eds Taylor C., Morgan K.,, pp. 139-167, vol. 1, 1980
 54. Spalding, D.B., "A General Purpose Computer Program for Multi-Dimensional, One and Two Phase Flow", Report HTS/81/1 Mech. Eng. Imperial College, London, 1981
 55. Rivard, W.W. and Torrey, M.D., "KFIX: A Program for Transient Two Dimensional Two Fluid Flow", Report LA-NUREG-6623, 1978
 56. Amsden, A.A., Harlow F.H., "KACHINA: An Eulerian Computer Program for Multifield Flows", Report LA-NUREG-5680, 1975
 57. Amsden, A.A., Harlow F.H., "KTIFA Two-Fluid Computer Program for Down comer Flow Dynamics", Report LA-NUREG-6994, 1977
 58. Darwish, M., Moukalled, F., and Balu, S., "A Unified Formulation of the Segregated Class of

- Algorithms for Multi-Fluid Flow at all Speeds," 36th AIAA/ASME/SAE/ASEE Joint Propulsion Conference, Huntsville, Alabama, 16-19 July, 2000.
59. Darwish, M., Moukalled, F., and, Sekar, B." A Unified Formulation of the Segregated Class of Algorithms for Multi-Fluid Flow at All Speeds," Numerical Heat Transfer, Part B: Fundamentals, (in print).
 60. Wesseling, P., "An introduction to Multigrid Methods," John Wiley & Sons Ltd., Baffins Lane, Chichester, West Sussex PO19 1UD, England, 1995.
 61. Brandt, A., Multilevel Computations: Review and Recent Developments. Multigrid Methods, S.F. McCormick (ed.) (Lecture Notes in Pure and Applied Mathematics 110) Marcel Dekker, New York, pp. 35-62, 1988.
 62. Dick, E., "Multigrid Formulation of polynomial flux-difference splitting for steady Euler Equations," Journal of Computational Physics, vol. 91, pp. 161-173, 1990.
 63. Dick, E. and Linden, J., "A Multigrid Flux-difference Splitting Method for Steady Incompressible Navier-Stokes Equations," Proc. 8th GAMM Conference on Numerical Methods in Fluid Mechanics, P. Wesseling (ed.) (Notes on Numerical Fluid Mechanics 29) Vieweg, Braunschweig, 1990.
 64. Hemker, P.W., "Defect Correction and Higher Order Schemes for the Multigrid Solution of the Steady Euler Equations," Multigrid Methods II, W. Hackbusch and U. Trottenberg (eds.) (Lecture Notes in Mathematics 1228) Springer, Berlin, pp. 149-165, 1986.
 65. Hemker, P.W. and Spekreijse, S.P., "Multiple Grid and Osher's Scheme for the Efficient solution of the Steady Euler Equations," Appl. Num. Math., vol. 2, pp. 475-493, 1986.
 66. Koren, B., "Defect Correction and Multigrid for an Efficient and Accurate Computation of Airfoil Flows," Journal of Computational Physics, vol. 77, pp. 183-206, 1988.
 67. Baldwin, B.S. and Lomax, H., "Thin Layer Approximation and algebraic model for separated turbulent flows," AIAA Paper 78-257, 1978.
 68. Sotiropoulos, F. and Patel, V.C., "Application of Reynolds-stress Transport Models to Stern and Wake Flow," J. Ship Res., vol. 39, p. 263, 1995.
 69. Cokljat, D., Ivanov, V.A., Srasola, F.J., and Vasquez, S.A., "Multiphase K-Epsilon Models for Unstructured Meshes," ASME 2000 Fluids Engineering Division Summer Meeting, June 11-15, 2000, Boston Massachusetts, USA.
 70. Pourahmadi, F. and Humphrey, J.A.C., "Modeling solid-fluid turbulent flows with application to predicting erosive wear," Int. J. Phys. Chem. Hydro, vol. 4, pp. 191-219, 1983.
 71. Elghobashi, S.E. and Abou-Arab, T.W., "A two-equation turbulence model for two-phase flows," Phys. Fluids, vo. 26, no. 4, pp. 931-938, 1983.
 72. Chen, C.P. and Wood, P.E., "Turbulence closure modeling of the dilute gas-particle axisymmetric jet," A.I.C.H.E Journal, vol. 32, no. 1, pp. 163-166, 1986.
 73. Mostafa, A.A. and Mongia, H.C., (1988) should be found
 74. Lopez de Bertodano, M., Lee, S.J., Lahey, R.T. Jr., and Drew, D.A., "The prediction of two-phase trubulence and phase distribution phenomena using a Reynolds stress model," ASME Journal of Fluids Engineering, vo. 112, pp. 107-113, 1990.
 75. Lopez de Bertodano, M., Lahey, R.T. Jr., and Jones, O.C., "Development of a k- ϵ model for bubbly two-phase flow," ASME Journal of Fluids Engineering, vo. 116, pp. 128-134, 1994.
 76. Lopez de Bertodano, M., Lahey, R.T. Jr., and Jones, O.C., "Phase distribution in bubbly two-phase flow in vertical ducts," Int. J. Multiphase flow, vol. 20, no. 5, pp. 805-818, 1994.
 77. Zwart, P.J., Raithby, G.D., and Raw, M.J., "An integrated Space-Time Finite-Volume Method for Moving-Boundary Problems," Numerical Heat Transfer Part B, vol. 34, pp. 257-270, 1998.

78. Spalding, D.B. and Markatos, N.C., "Computer Simulation of Multi-Phase Flows: A Course of Lectures and Computer Workshops", Report CFD/83/4, Mech. Eng., Imperial College, London, 1983.
79. Lo, S.M., "Multiphase Flow Model in harwell-FLOW#D Computer Code", Report AEA-InTec-0062, 1990.
80. Morsi, S.A. and Alexander, A.J., "An investigation of Particle Trajectories in Two-Phase Flow System," Journal of Fluid Mechanics, vol. 55, part 2, pp. 193-208, 1972.
81. Baghdadi, A.H.A., "Numerical Modelling of Two-Phase Flow With Inter-Phase Slip," Ph.D. Thesis, Imperial College, University of London, 1979.
82. Rubin S.G., Khosla P.K., Polynomial Interpolation Method for Viscous Flow Calculations, J. Comput. Phys., vol. 27, pp. 153-168, 1982.
83. Serizawa, A., Kataoka, I., and Michiyoshi, I., "Phase Distribution in Bubbly Flow," Data set No. 24, Proceedings of the Second International Workshop on Two-Phase Flow Fundamentals, Rensselaer Polytechnic Institute, Troy, NY, 1986.
84. Wang, S.k., Lee, S-j. Jones, Jr., O.C., and Lahey, Jr., R.T., "3-D Turbulence structure and Phase Distribution Measurements in Bubbly Two-Phase Flows," International Journal of Multiphase Flow, vol. 13, No. 3, 1987.
85. Antal, S.P., Lahey,R.T., and Flaherty, J.E., "Analysis of Phase Distribution in Fully Developed Laminar Bubbly Two-Phase Flows," International Journal of Multiphase Flow, vol. 17, No. 5, pp. 635-652, 1991.
86. Sato, Y., sadatomi, M., and Sekoguchi, K., "Momentum and Heat Transfer in Two-Phase Bubble flow," International Journal of Multiphase Flow, I. Theory, pp. 167-177, II. A Comparison Between Experiment and Theoretical Calculations, pp. 179-190, 1981.
87. Lopez de Bertodano, M., Lee, S-J., Lahey, Jr., R.T., and Drew, D.A., "The Prediction of Two-Phase Turbulence and Phase Distribution Phenomena Using a Reynolds Stress Model," Journal of Fluids Engineering, vo. 112, pp. 107-113, 1990.
88. Lopez de Bertodano, M., Lahey, Jr., R.T., and Jones, O.C., "Development of a k- ϵ Model for Bubbly Two-Phase Flow," Journal of Fluids Engineering, vo. 116, pp. 128-134, 1994.
89. Lopez de Bertodano, M., Lahey, Jr., R.T., and Jones, O.C., "Phase Distribution in Bubbly Two-Phase Flow in Vertical Ducts," International Journal of Multiphase Flow, vol. 20, No. 5, pp. 805-818, 1994.
90. Nakoryakov, V.E., Kashinsky, O.N., Randin, V.V., and Timkin, L.S., "Gas-Liquid Bubbly Flow in Vertical Pipes," Journal of Fluids Engineering, vo. 118, pp. 377-382, 1996.
91. Boisson, N. and Malin, M.R., "Numerical Prediction of Two-Phase Flow in Bubble Columns," International Journal for Numerical Methods in Fluids, vol. 23, pp. 1289-1310, 1996.
92. Huang, B., "Modelisation Numerique D'ecoulements Diphasiques a Bulles dans les Reacteurs Chimiques," Ph.D. Thesis, Universite Claude Bernard, Lyon, 1989.
93. Peterson, K.O., "Etude Experimentale et Numerique des Ecoulements Diphasiques dans les Reacteurs Chimiques," Ph.D. Thesis, Universite Claude Bernard, Lyon, 1992.
94. Lahey, R.T., Lopez de Bertodano, M., and Jones, O.C., "Phase Distribution in Complex Geometry Ducts," Nuclear Engineering Design, vol. 141, p. 177, 1993.
95. Drew, D.A. and Lahey, Jr., T.J., "The Virtual Mass and Lift Force on a Sphere in Rotating and Straining Inviscid Flow," International Journal of Multiphase Flow, vol. 13, No. 1, p. 113, 1987.
96. PHOENICS: "http://www.cham.co.uk/phoenics/d_polis/d_applic/d_flows/laheya.htm"
97. Tsuji, Y., Morikawa, Y., and Shiomi, H., "LDV Measurements of an Air-Solid Two-Phase Flow

- in a Vertical Pipe," *Journal of Fluid Mechanics*, vo. 139, pp. 417-434, 1984.
98. Adeniji-Fashola, A. and Chen, C.P., "Modeling of Confined Turbulent Fluid-Particle Flows Using Eulerian and Lagrangian Schemes," *International Journal of Heat and Mass transfer*, vol.33, pp. 691-701, 1990.
99. Naik, S. and Bryden, I.G., "Prediction of Turbulent Gas-Solids Flow in Curved Ducts Using The Eulerian-Lagrangian Method," *International Journal for Numerical Methods in Fluids*, vol. 31, pp. 579-600, 1999.
100. Harlow, F.H. and Amsden, A.A., "Numerical Calculation of Multiphase Fluid Flow," *Journal of Computational Physics*, vol. 17, pp. 19-52, 1975.
101. Osiptsov, A.N., "Structure of the Laminar Boundary Layer of a Disperse Medium on a Flat Plate," *Fluid Dyn.*, vol. 15, pp. 512-517, 1980.
102. Prabha, S. and Jain, A.C., "On the Use of Compatibility Conditions in the Solution of Gas Particulate Boundary Layer Equations," *Appl. Sci. res.*, vol. 36, pp. 81-91, 1980.
103. Sgleton, R.E., "The Incompressible Gas Solid Particle Flows Over a Semi-Infinite Flat Plate," *Z. Angew Math. Phys.*, vol. 19, p. 545, 1965.
104. Wang, B.Y., and Glass, I.I., "Compressible Laminar Boundary Layer Flows of a Dusty Gas Over a Semi-Infinite Flat Plate," *Journal of Fluid Mechanics*, vol. 186, pp. 223-241, 1988.
105. Soo, S.L., "Boundary Layer Motion of a Gas-Solids Suspension," *Proc. Symp. Interaction Between Fluids and Particles, Inst. Chem. Eng.*, pp. 50-63, 1962.
106. Chamkha, A.J., and Peddeson, J.J.R., "Boundary Layer Flow of a Particulate Suspenson Past a Flat Plate," *International Journal of Multiphase Flow*, vol. 17, pp. 805-808, 1991.
107. Thevand, N., Daniel, E., and Loraud, J.C., "On high-Resolution Schemes for Solving Unsteady Compressible Two-Phase Dilute Viscous Flows," *International Journal of Numerical Methods in Fluids*, vol. 31, pp. 681-702, 1999.
108. Chang, I.S., "One and Two-Phase Nozzle Flows," *AIAA J.*, vol. 18, pp. 1455-1461, 1980.
109. Ishii, R. and Kawasaki, K., "Limiting Particle Streamline in the flow of a Gas-Particle Mixture Through an Axially Symmetric Nozzle," *Phys. Fluids*, vol. 25, No. 6, pp. 959-966, 1982.
110. Ishii, R., Umeda, Y., and Kawasaki, K., "Nozzle Flows of Gas-Particle Mixtures," *Phys. Fluids*, vol. 30, No. 3, pp. 752-760, 1987.
111. Hwang, C.J. and Chang, G.C., "Numerical Study of Gas-Particle Flow in a Solid Rocket Nozzle," *AIAA Journal*, vol. 26, No. 6, pp. 682-689, 1988.
112. Chang, H.T., Hourng, L.W., and Chien, L.E., "Application of Flux-Vector-Splitting Scheme to a Dilute Gas-Particle JPL Nozzle Flow," *International Journal for Numerical Methods in Fluids*, vol. 22, pp. 921-935, 1996.
113. Mehta, R.C. and Jayachandran, T., "A Fast Algorithm to Solve Viscous Two-Phase Flow in an Axisymmetric Rocket Nozzle," *International Journal for Numerical Methods in Fluids*, vol. 26, pp. 501-517, 1998.
114. Igra, O., Elperin, I., and Ben-Dor, G., "Dusty Gas Flow in a Converging-Diverging Nozzle," *Journal of Fluids Engineering*, vol. 121, pp. 908-913, 1999.
115. Back, L.H. and Cuffel, R.F., "Detection of Oblique Shocks in a Conical Nozzle with a Circular-arc Throat," *AIAA Journal*, vol. 4, pp. 2219-2221, 1966.
116. Back, L.H. Massier, P.F., and Cuffel, R.F., "Flow Phenomena and Convective Heat Transfer in a Conical Supersonic Nozzle," *Journal of Spacecraft*, vol. 4, pp. 1040-1047, 1967.
117. Cuffel, R.F., Back, L.H., and Massier, P.F., "Transonic Flowfield in a Supersonic Nozzle with Small Throat Radius of Curvature," *AIAA Journal*, vol. 7, pp. 1364-1366, 1969.

List of Table

Table 1 Normalized CPU-times for turbulent bubbly flow in a pipe.

Table 2 Normalized CPU-times for turbulent air-particle flow in a pipe.

Table 3 Normalized CPU-times for Dusty flow over a flat plate.

Table 3 Normalized CPU-times for Dusty flow in a converging-diverging nozzle.

Figure Captions

Fig. 1 (a) Control volume, (b) Typical control volume faces and geometric nomenclature.

Fig. 2 Physical domain for the gas-particle transport problem.

Fig. 3 (a) Comparison between the analytical and numerical particle velocity distributions, (b)-(g) convergence histories on the different grid systems, (h) and convergence histories of the various algorithms on the 80 C.V. grid for the horizontal dilute gas-solid flow problem.

Fig. 4 Normalized CPU-times for the horizontal dilute gas-solid flow problem.

Fig. 5 (a) gas and particle velocity distributions, (b)-(g) convergence histories on the different grid systems, (h) and convergence histories of the various algorithms on the 80 C.V. grid for the horizontal dense gas-solid flow problem.

Fig. 6 Normalized CPU-times for the horizontal dense gas-solid flow problem.

Fig. 7 (a) Liquid and gas velocity distributions, (b)-(g) convergence histories on the different grid systems, (h) and convergence histories of the various algorithms on the 80 C.V. grid for the horizontal dilute bubbly flow problem.

Fig. 8 Normalized CPU-times for the horizontal dilute bubbly flow problem.

Fig. 9 (a) Liquid and gas velocity distributions, (b)-(g) convergence histories on the different grid systems, (h) and convergence histories of the various algorithms on the 80 C.V. grid for the horizontal dense bubbly flow problem.

Fig. 10 Normalized CPU-times for the horizontal dense bubbly flow problem.

Fig. 11 (a) gas and particle velocity distributions, and (b)-(h) convergence histories on the different grid systems for the vertical dilute gas-solid flow problem.

Fig. 12 Normalized CPU-times for the vertical dilute gas-solid flow problem.

Fig. 13 (a) gas and particle velocity distributions, and (b)-(h) convergence histories on the different grid systems for the vertical dense gas-solid flow problem.

Fig. 14 Normalized CPU-times for the vertical dense gas-solid flow problem.

Fig. 15 (a) gas and particle velocity distributions; and (b)-(h) convergence histories on the different grid systems for the vertical dilute bubbly flow problem.

Fig. 16 Normalized CPU-times for the vertical dilute bubbly flow problem.

Fig. 17 (a) gas and particle velocity distributions, and (b)-(h) convergence histories on the different grid systems for the vertical dense bubbly flow problem.

Fig. 18 Normalized CPU-times for the vertical dense bubbly flow problem.

Fig. 19 Comparison of fully developed liquid velocity and void fraction profiles for turbulent bubbly upward bubbly flow in a pipe; (a) Seriwaza et al. data, (b) Lahey et al. data.

Fig. 20 (a)-(g) Convergence histories of the SG, PG, and FMG methods on the finest grid, and (h) convergence histories of the various algorithms on the finest mesh using the FMG method for turbulent upward bubbly flow in a pipe.

Fig. 21 Comparison of fully developed gas and particle velocity profiles for turbulent air-particle flow in a pipe.

Fig. 22 Convergence histories of the (a) SIMPLE, (b) SIMPLEC, (c) SIMPLEST, and (d) SIMPLEX algorithms using the SG, PG, and FMG methods on the finest mesh for turbulent air-particle flow in a pipe.

Fig. 23 The three different regions within the boundary layer of dusty flow over a flat plate.

Fig. 24 Comparison of fully developed gas and particle velocity profiles inside the boundary layer at different axial locations for dilute two-phase flow over a flat plate

Fig. 25 (a)-(g) Convergence histories of the SG, PG, and FMG methods on the finest grid, and (h) convergence histories of the various algorithms on the finest mesh using the FMG method for dusty gas flow over a flat plate.

Fig. 26 Physical domain for the dusty gas flow in a converging-diverging nozzle.

Fig. 27 (a,b) Volume Fraction contours and (c,d) particle velocity vectors for dusty gas flow in a converging-diverging nozzle.

Fig. 28 Comparison of one-phase and two-phase gas Mach number distributions along the (a) wall and (b) centerline of the dusty flow in a converging-diverging nozzle problem.

Fig. 29 Convergence histories of the (a) SIMPLE, (b) SIMPLEC, (c) SIMPLEST, and (d) SIMPLEX algorithms using the SG method for dusty gas flow in a converging-diverging nozzle.

Table 1 Normalized CPU-times for turbulent bubbly flow in a pipe.

GRID	METHOD	ALGORITHMS					
		SIMPLE	SIMPLEX	SIMPLEST	SIMPLEX	PISO	PRIME
12x4 C.V.	SG	1.00	0.97	0.98	1.02	1.27	0.91
	SG	3.30	3.43	3.42	3.79	5.54	3.77
	PG (2 levels)	3.62	3.53	3.58	3.97	5.70	3.87
	SG/PG	0.91	0.97	0.95	0.96	0.97	0.97
	FMG (2 levels)	3.10	3.04	3.12	4.68	5.34	8.81
	SG/FMG	1.07	1.13	1.10	0.81	1.04	0.43
24x8 C.V.	SG	25.34	25.71	26.19	30.52	49.17	33.43
	PG (3 levels)	21.92	22.37	23.36	24.27	44.43	70.17
	SG/PG	1.16	1.15	1.12	1.26	1.11	0.48
	FMG (3 levels)	14.44	13.55	14.23	21.74	22.02	45.61
	SG/FMG	1.75	1.90	1.84	1.40	2.23	0.73
	SG	1231.38	1238.38	1296.28	1249.46	2038.94	1934.27
48x16 C.V.	PG (4 levels)	978.29	978.84	1037.55	1002.88	1593.90	1730.55
	SG/PG	1.26	1.27	1.25	1.25	1.28	1.12
	FMG (4 levels)	207.61	208.46	217.84	214.60	836.08	352.91
	SG/FMG	5.93	5.94	5.95	5.82	2.44	5.48
96x32 C.V.							6.55

Table 2 Normalized CPU-times for turbulent air-particle flow in a pipe.

		ALGORITHMS				
GRID	METHOD	SIMPLE	SIMPLEC	SIMPLEX	SIMPLEST	
12x5 C.V.	SG	1.00	0.97	0.99	1.36	
	SG	13.57	13.64	14.25	13.88	
	PG (2 levels)	7.36	7.28	7.50	7.58	
	SG/PG	1.84	1.87	1.90	1.83	
	FMG (2 levels)	6.88	6.14	7.18	8.81	
	SG/FMG	1.97	2.22	1.98	1.58	
24x10 C.V.	SG	218.51	217.47	217.41	213.16	
	PG (3 levels)	26.30	26.49	27.79	27.78	
	SG/PG	8.31	8.21	7.82	7.67	
	FMG (3 levels)	22.15	21.21	22.22	22.95	
	SG/FMG	9.87	10.25	9.78	9.29	
	SG	1503.37	1506.72	1593.35	1525.79	
48x20 C.V.	PG (4 levels)	405.18	409.34	433.50	432.55	
	SG/PG	3.71	3.68	3.68	3.53	
	FMG (4 levels)	163.73	164.61	172.39	170.69	
	SG/FMG	9.18	9.15	9.24	8.94	

Table 3 Normalized CPU-times for Dusty flow over a flat plate.

		ALGORITHMS						
GRID	METHOD	SIMPLEX	SIMPLEXC	SIMPLEX	SIMPLEST	SIMPLEXM	PISO	PRIME
13x6 C.V.	SG	1.00	0.96	0.99	1.06	1.43	0.74	1.12
	SG	2.79	3.01	3.13	3.65	3.74	2.64	3.95
	PG (2 levels)	3.15	3.23	3.12	4.67	3.75	2.52	4.57
	SG/PG	0.89	0.93	1.00	0.78	1.00	1.05	0.87
	FMG (2 levels)	2.63	2.50	2.53	8.85	3.54	3.11	7.50
	SG/FMG	1.06	1.20	1.23	0.41	1.06	0.85	0.53
26x12 C.V.	SG	21.77	21.83	23.30	30.87	28.06	27.08	44.65
	PG (3 levels)	12.85	12.94	14.04	17.90	15.83	16.64	25.50
	SG/PG	1.69	1.69	1.66	1.72	1.77	1.63	1.75
	FMG (3 levels)	9.35	9.66	9.92	27.02	11.86	16.19	27.09
	SG/FMG	2.33	2.26	2.35	1.14	2.37	1.67	1.65
	SG	409.28	411.61	423.36	784.67	464.81	543.69	1421.28
52x24 C.V.	PG (4 levels)	309.12	307.29	315.81	693.46	331.59	466.57	749.27
	SG/PG	1.32	1.34	1.34	1.13	1.40	1.17	1.90
	FMG (4 levels)	57.46	51.09	53.47	177.52	65.36	116.98	187.08
	SG/FMG	7.12	8.06	7.92	4.42	7.11	4.65	7.60

Table 4 Normalized CPU-times for Dusty flow in a converging-diverging nozzle.

GRID	METHOD	ALGORITHMS		
		SIMPLE	SIMPLEX	SIMPLEST
47x20 C.V.	SG	1.00	0.97	1.06
94x40 C.V.	SG	7.36	7.20	7.77
188x80 C.V.	SG	54.39	55.22	60.12
				56.69

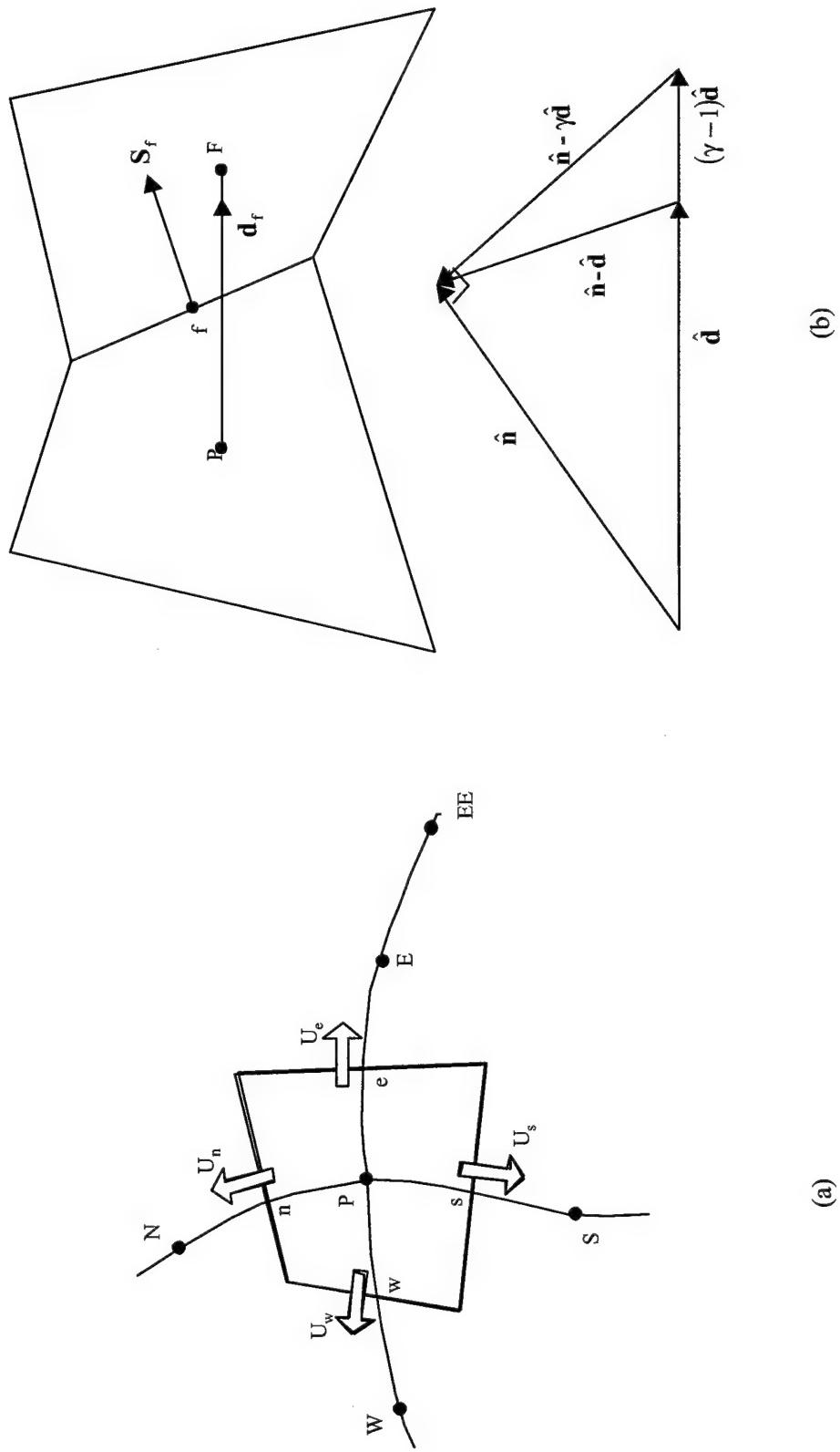


Fig. 1 (a) Control volume, (b) Typical control volume faces and geometric nomenclature.

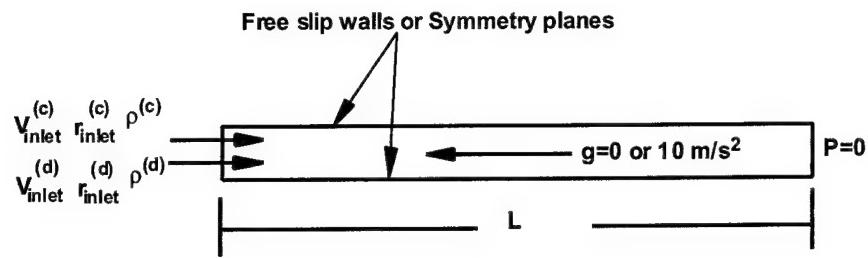


Fig. 2 Physical domain for the gas-particle transport problem.

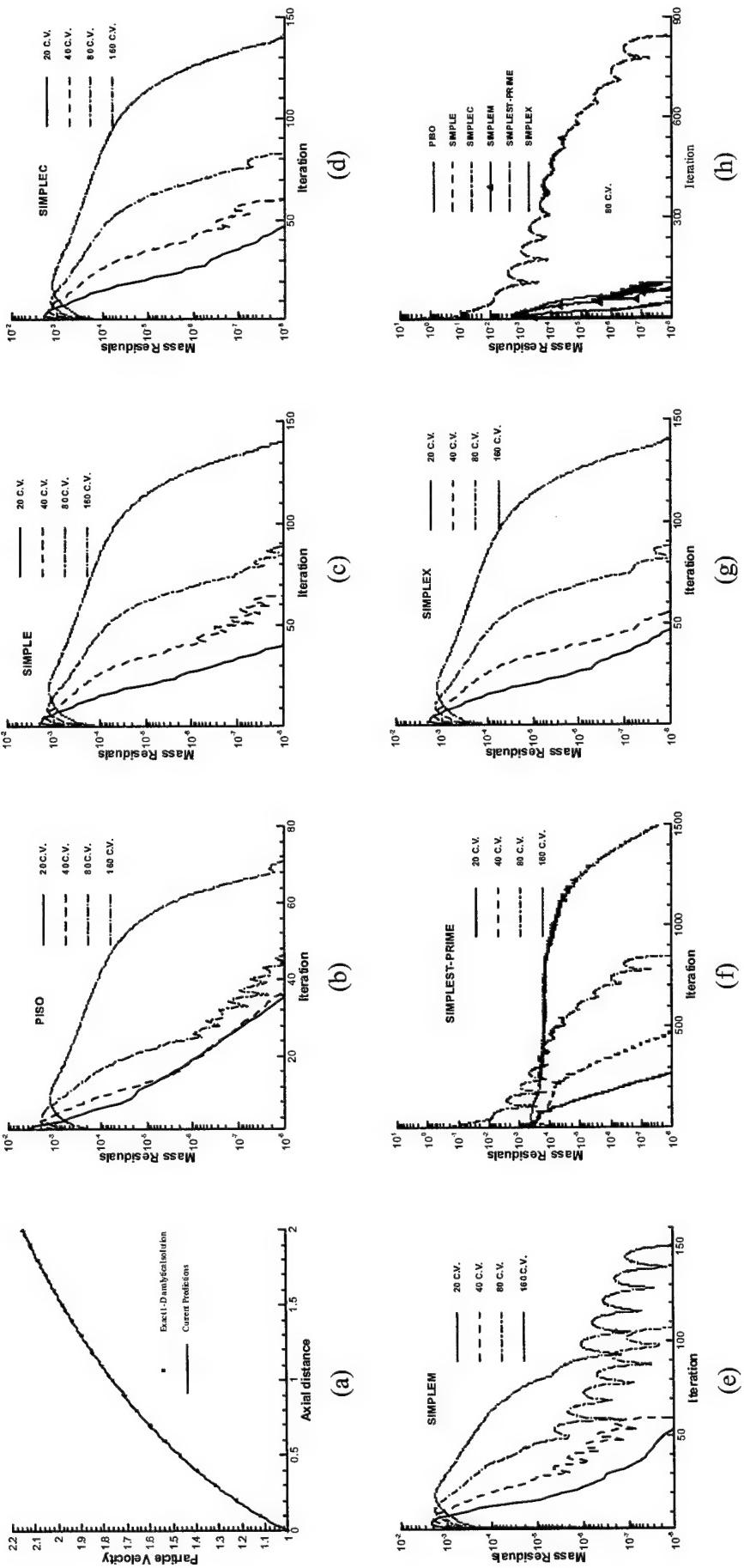


Fig. 3 (a) Comparison between the analytical and numerical particle velocity distributions, (b)-(g) convergence histories on the different grid systems, (h) and convergence histories of the various algorithms on the 80 C.V. grid for the horizontal dilute gas-solid flow problem.

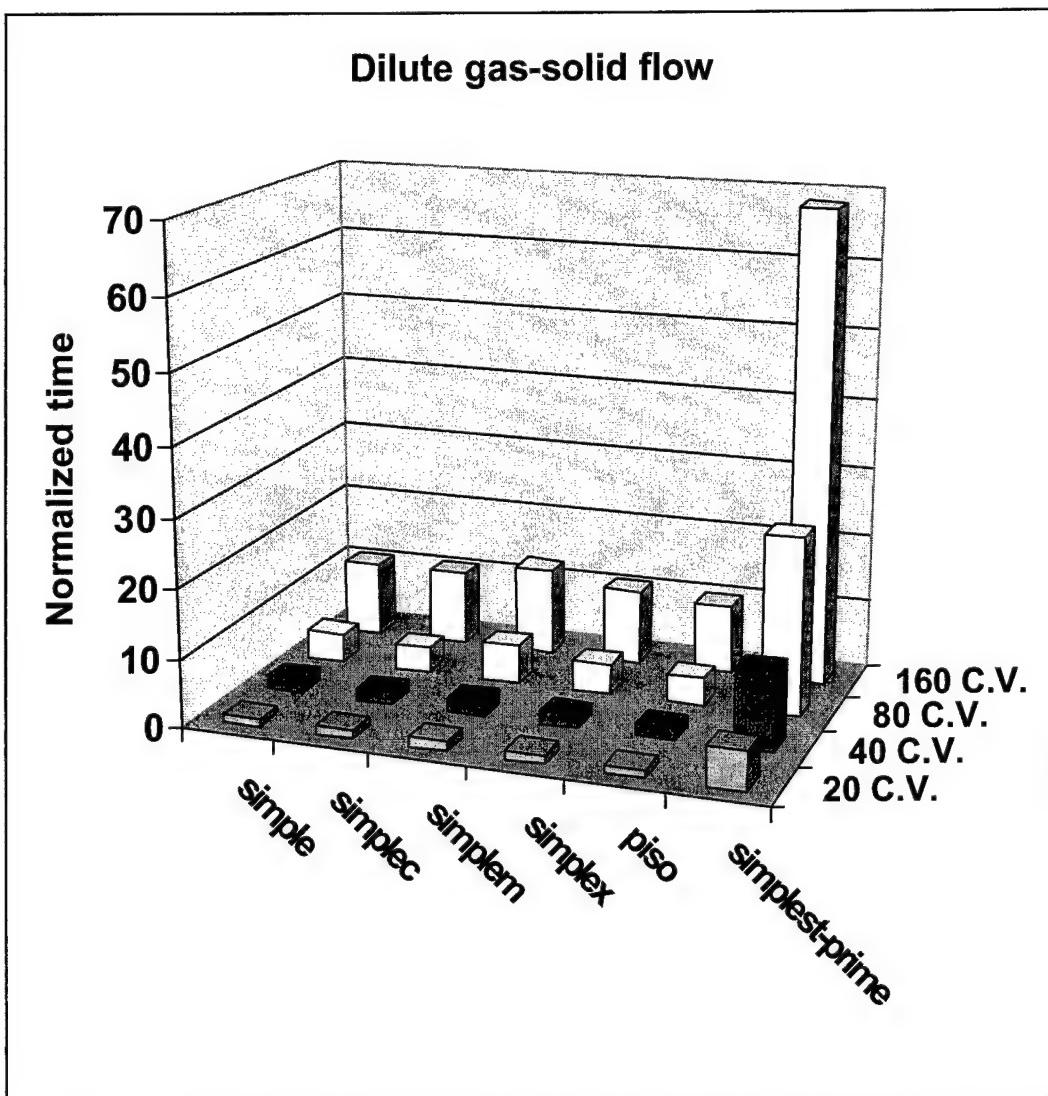


Fig. 4 Normalized CPU-times for the horizontal dilute gas-solid flow problem.

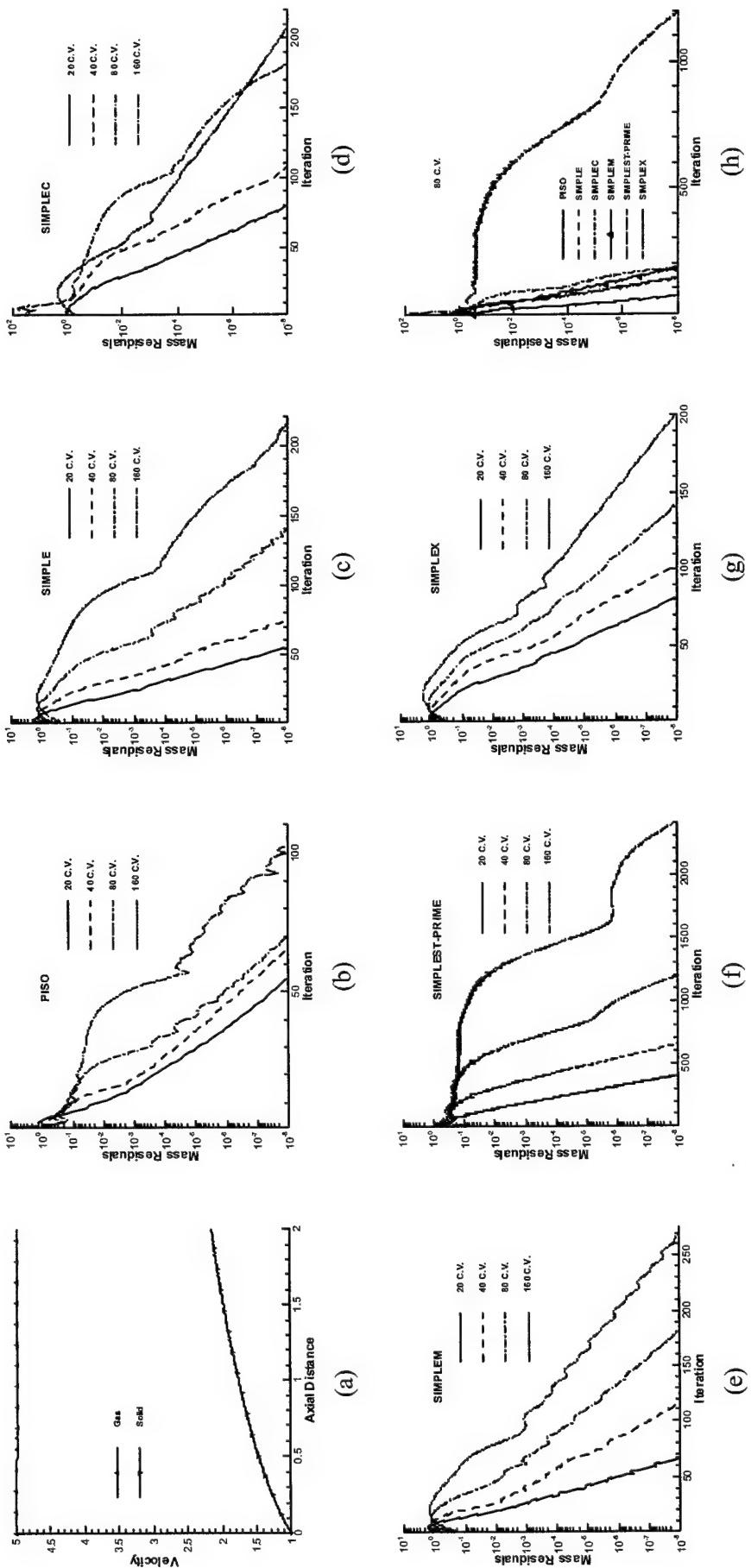


Fig. 5 (a) gas and particle velocity distributions, (b)-(g) convergence histories on the 80 C.V. grid for the horizontal dense gas-solid flow problem, and (h) and convergence histories of the various algorithms on the 80 C.V. grid for the horizontal dense gas-solid flow problem.

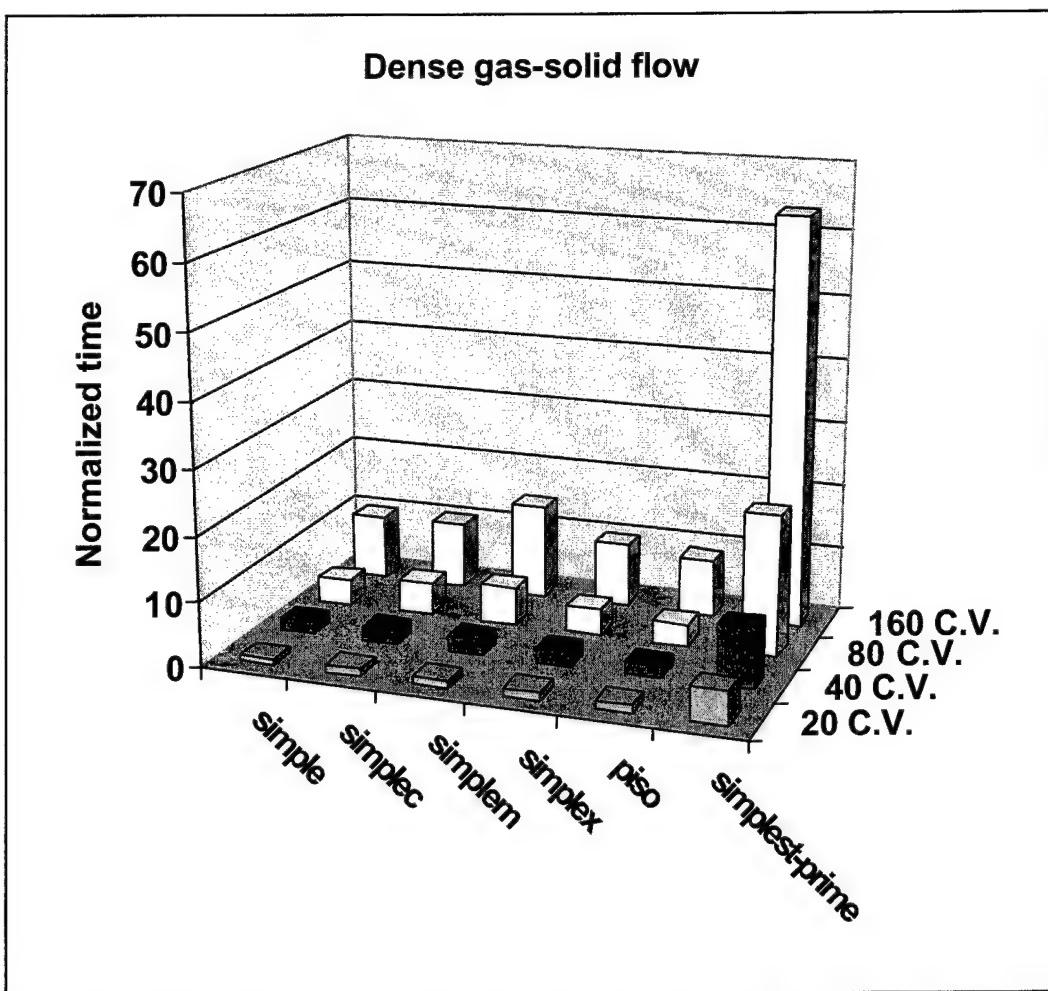


Fig. 6 Normalized CPU-times for the horizontal dense gas-solid flow problem.

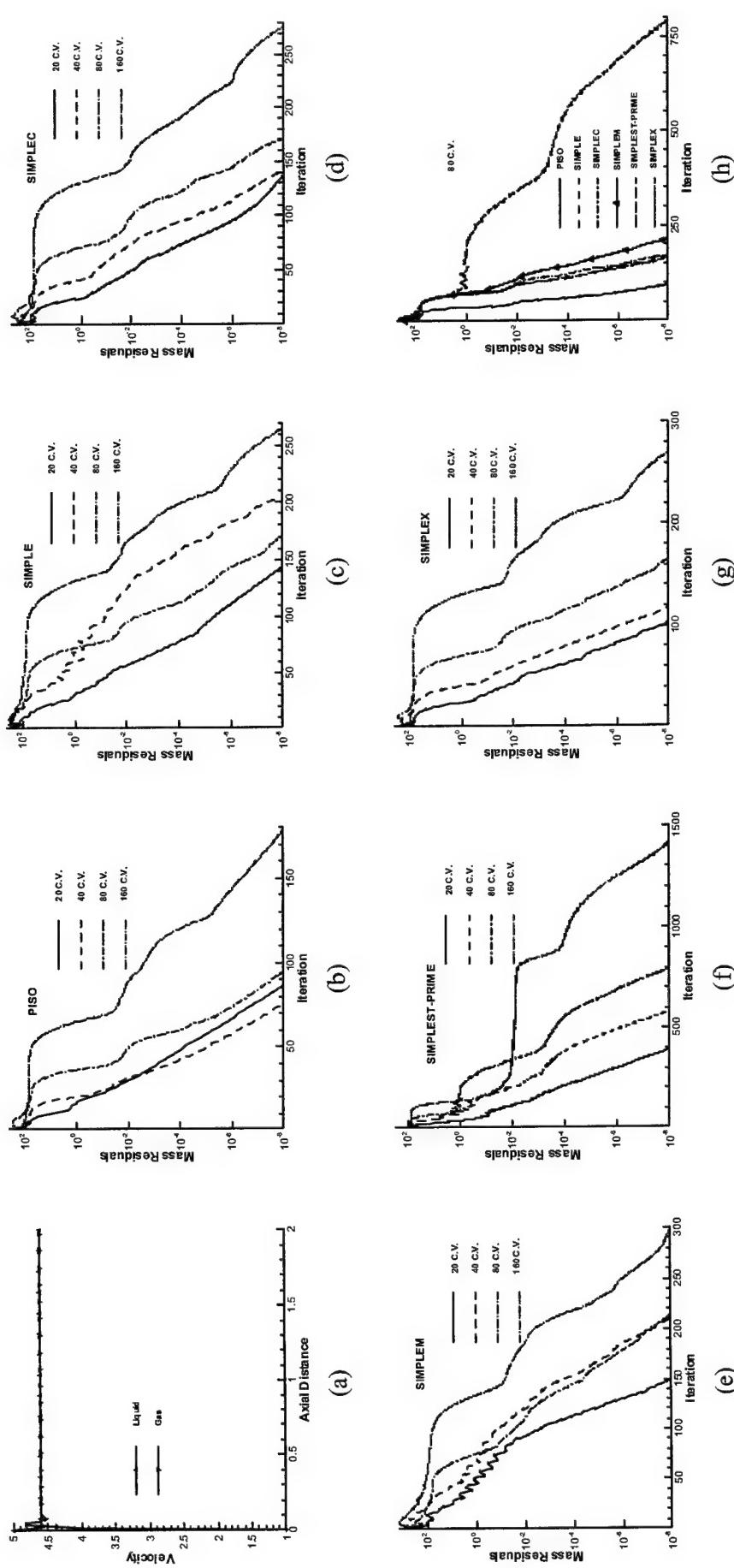


Fig. / (a) Liquid and gas velocity distributions, (b)-(g) convergence histories on the uniform grid systems, (h) and convergence histories on the various algorithms on the 80 C.V. grid for the horizontal dilute bubbly flow problem.

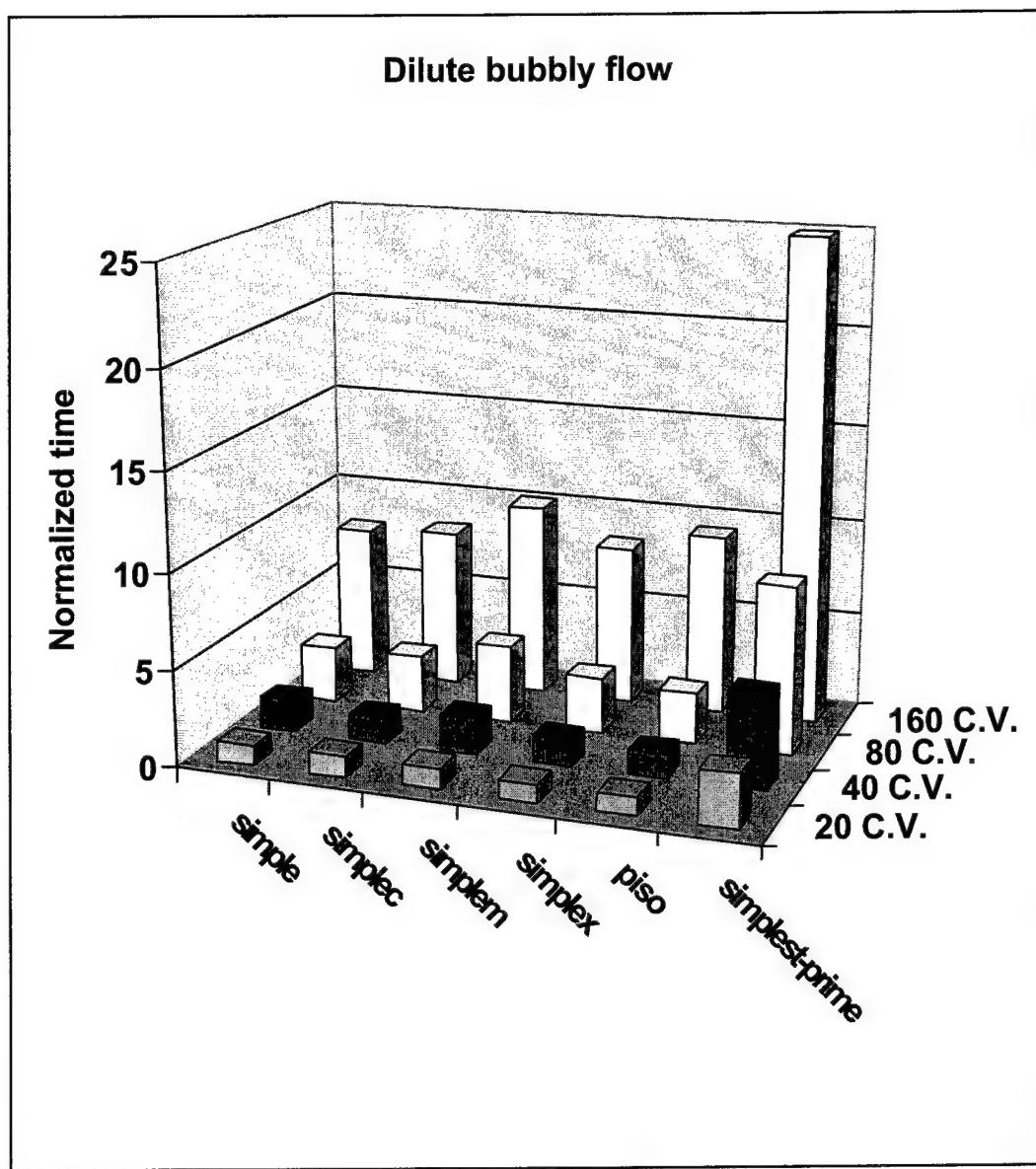


Fig. 8 Normalized CPU-times for the horizontal dilute bubbly flow problem.

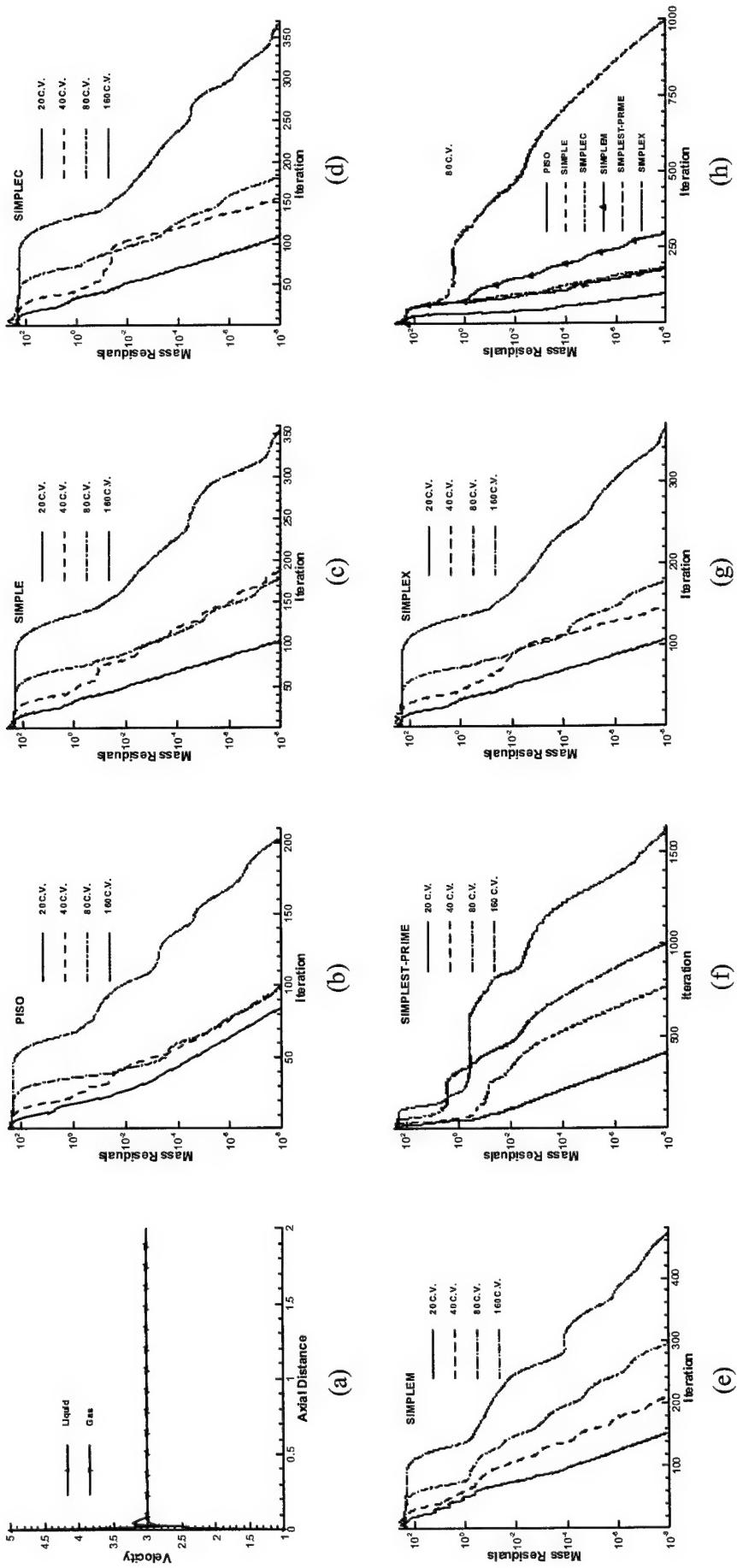


Fig. 9 (a) Liquid and gas velocity distributions, (b)-(g) convergence histories on the different grid systems, (h) and convergence histories of the various algorithms on the 80 C.V. grid for the horizontal dense bubbly flow problem.

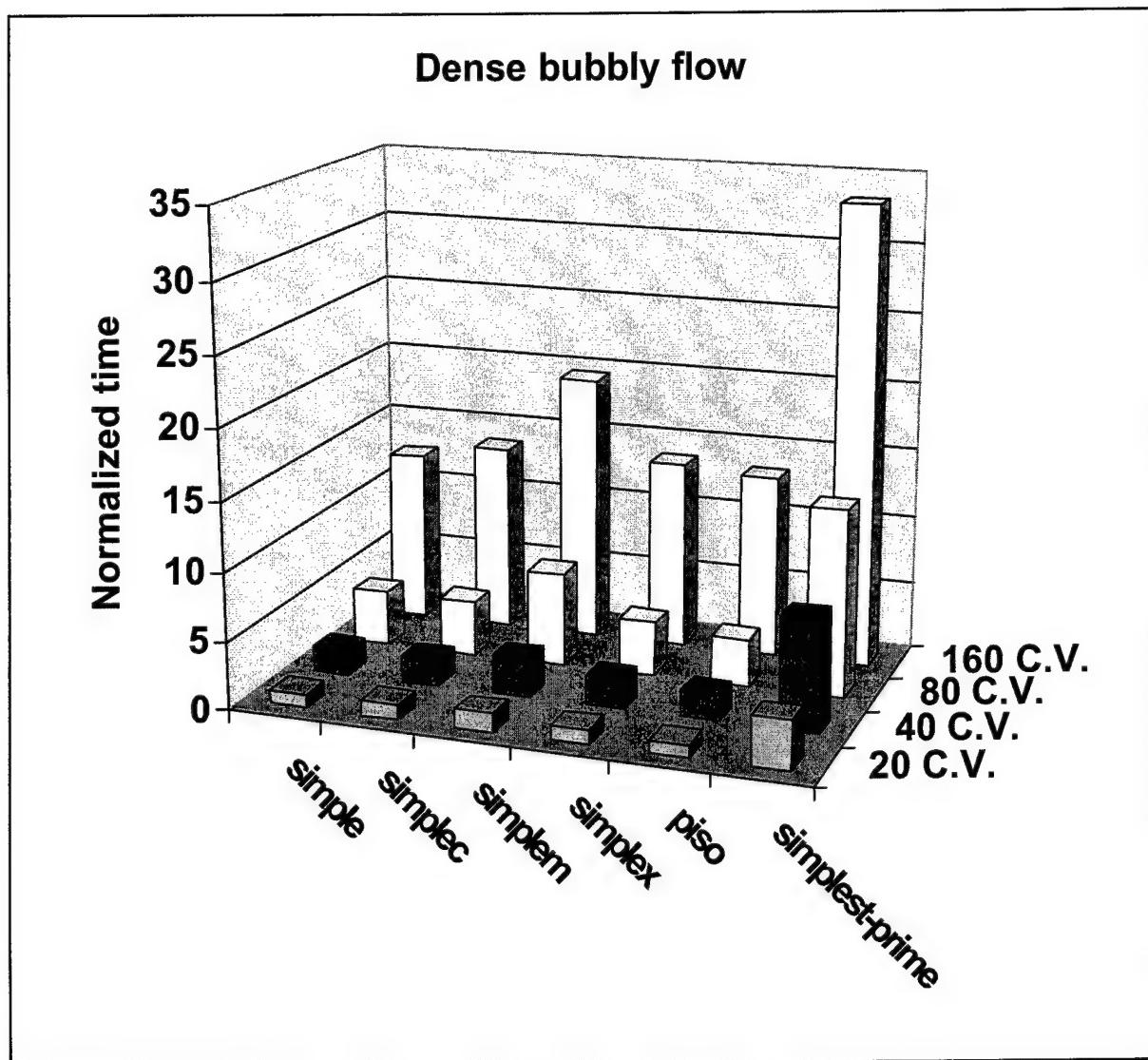


Fig. 10 Normalized CPU-times for the horizontal dense bubbly flow problem.

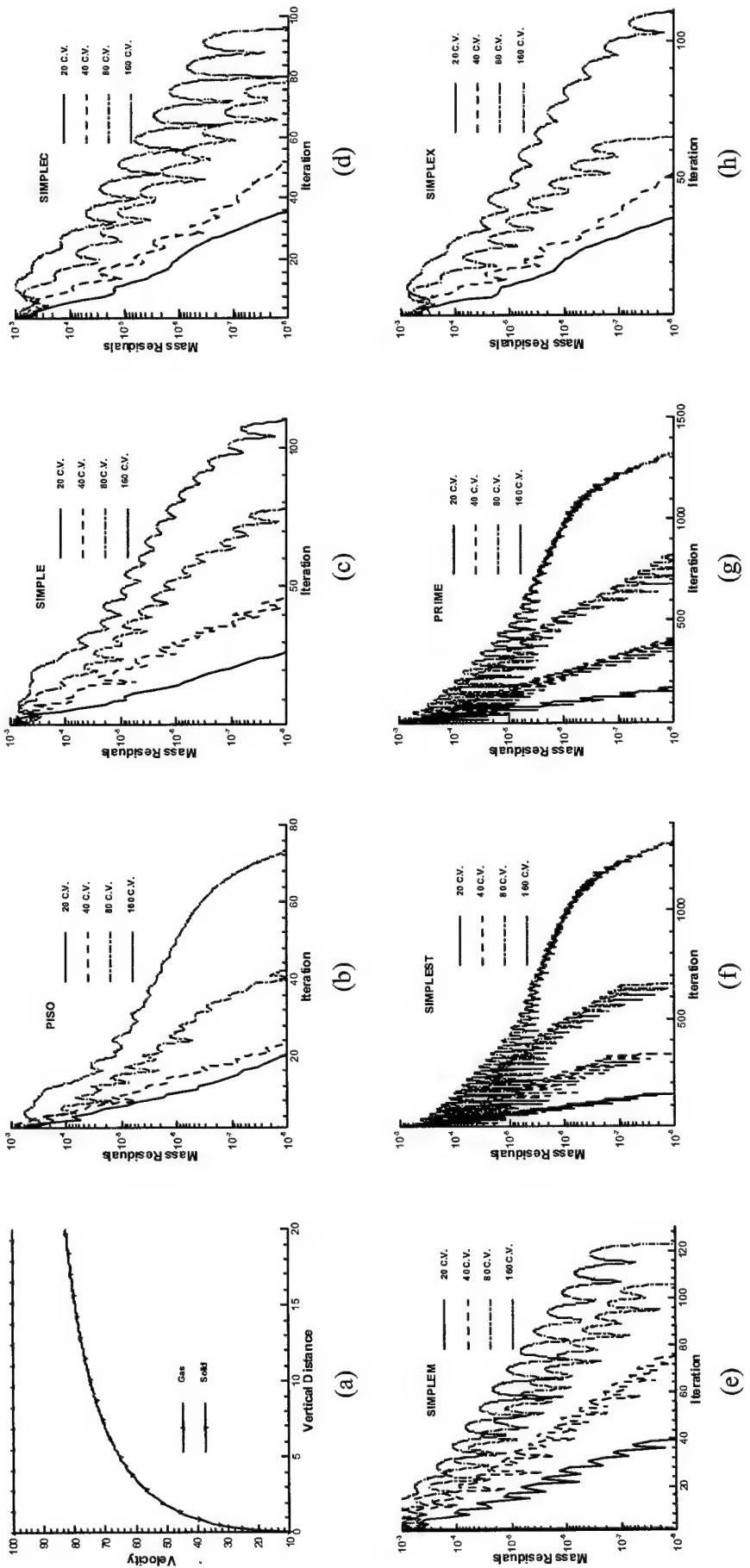


Fig. 11 (a) gas and particle velocity distributions, and (b)-(h) convergence histories on the different grid systems for the vertical dilute gassolid flow problem.

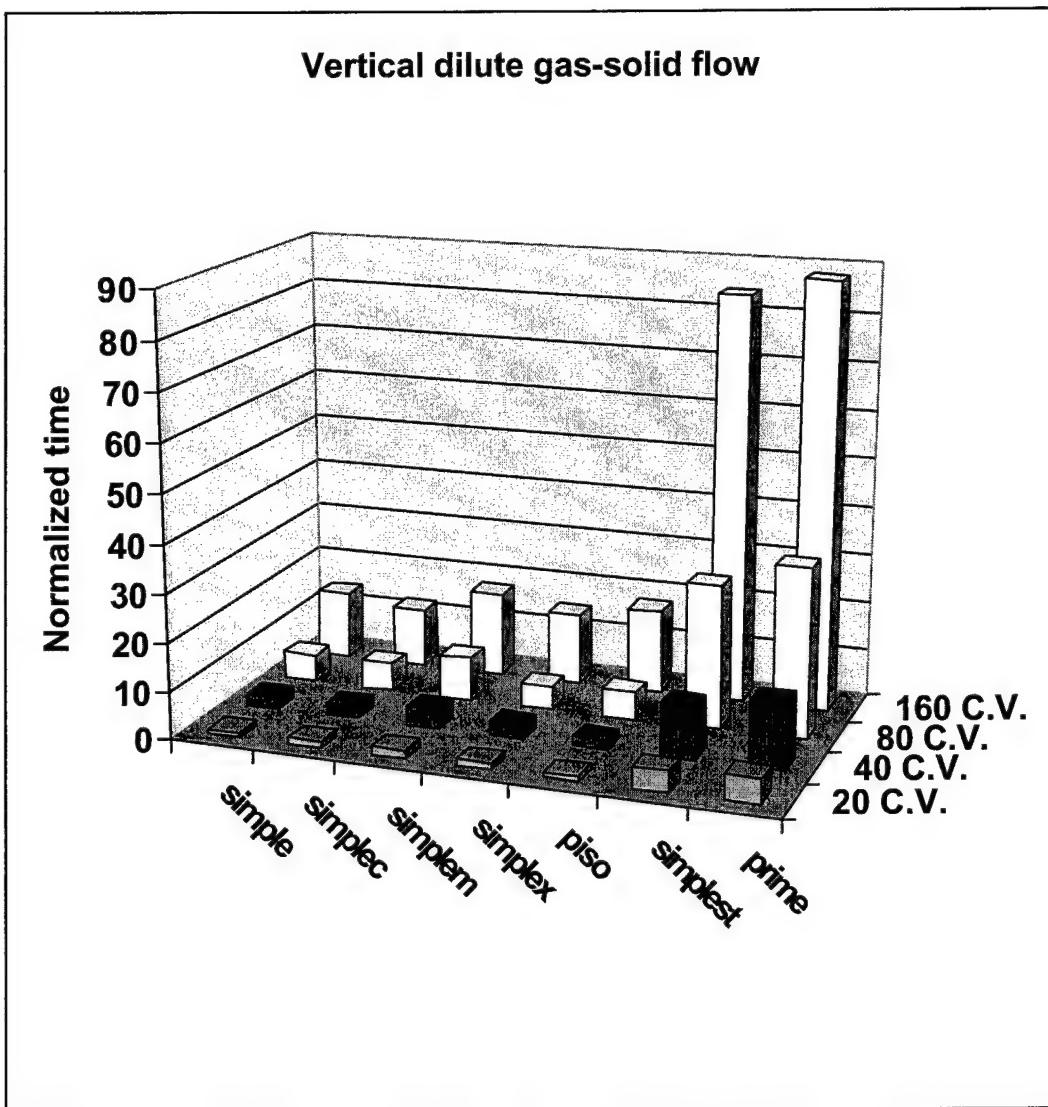


Fig. 12 Normalized CPU-times for the vertical dilute gas-solid flow problem.

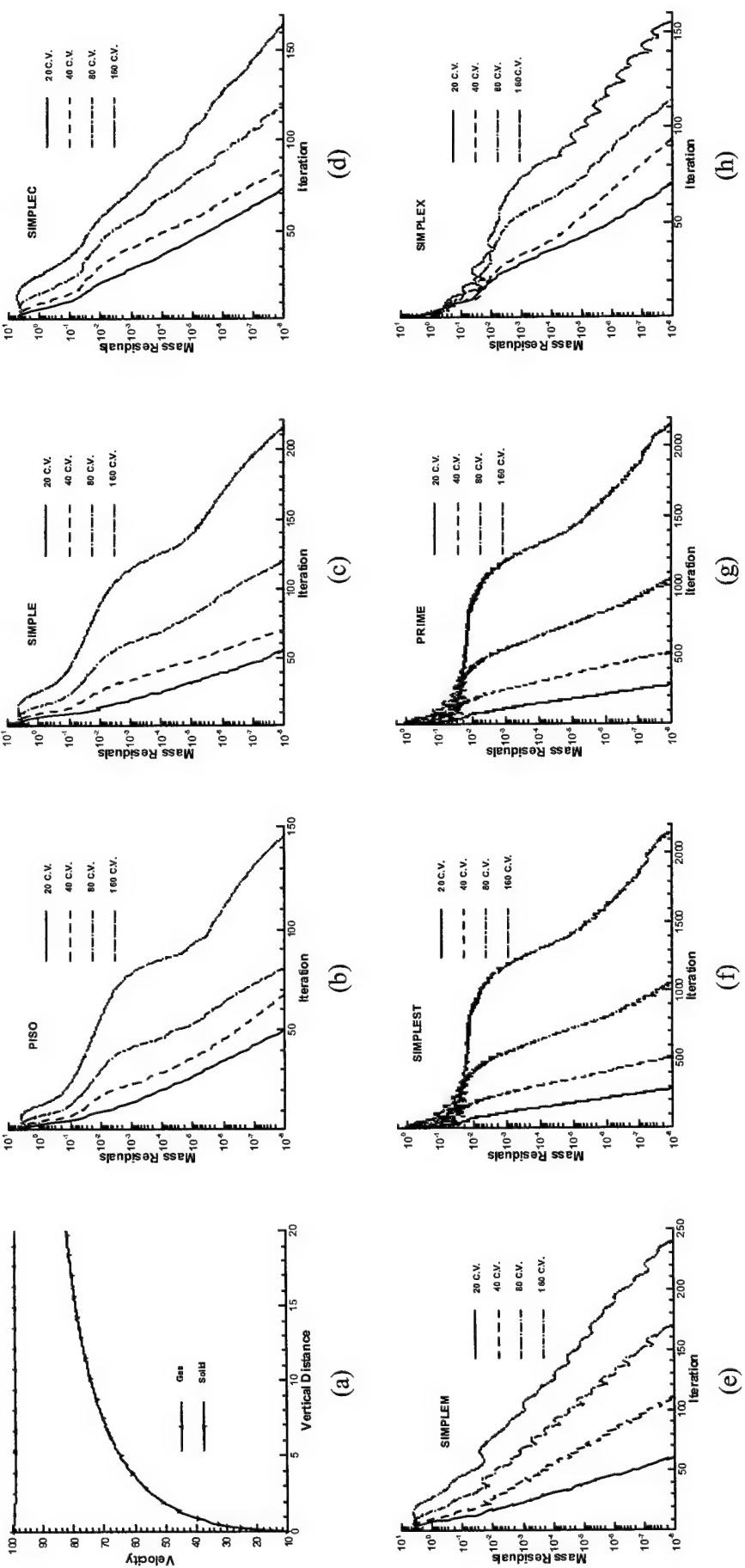


Fig. 13 (a) gas and particle velocity distributions, and (b)-(h) convergence histories on the different grid systems for the vertical dense gas-solid flow problem.

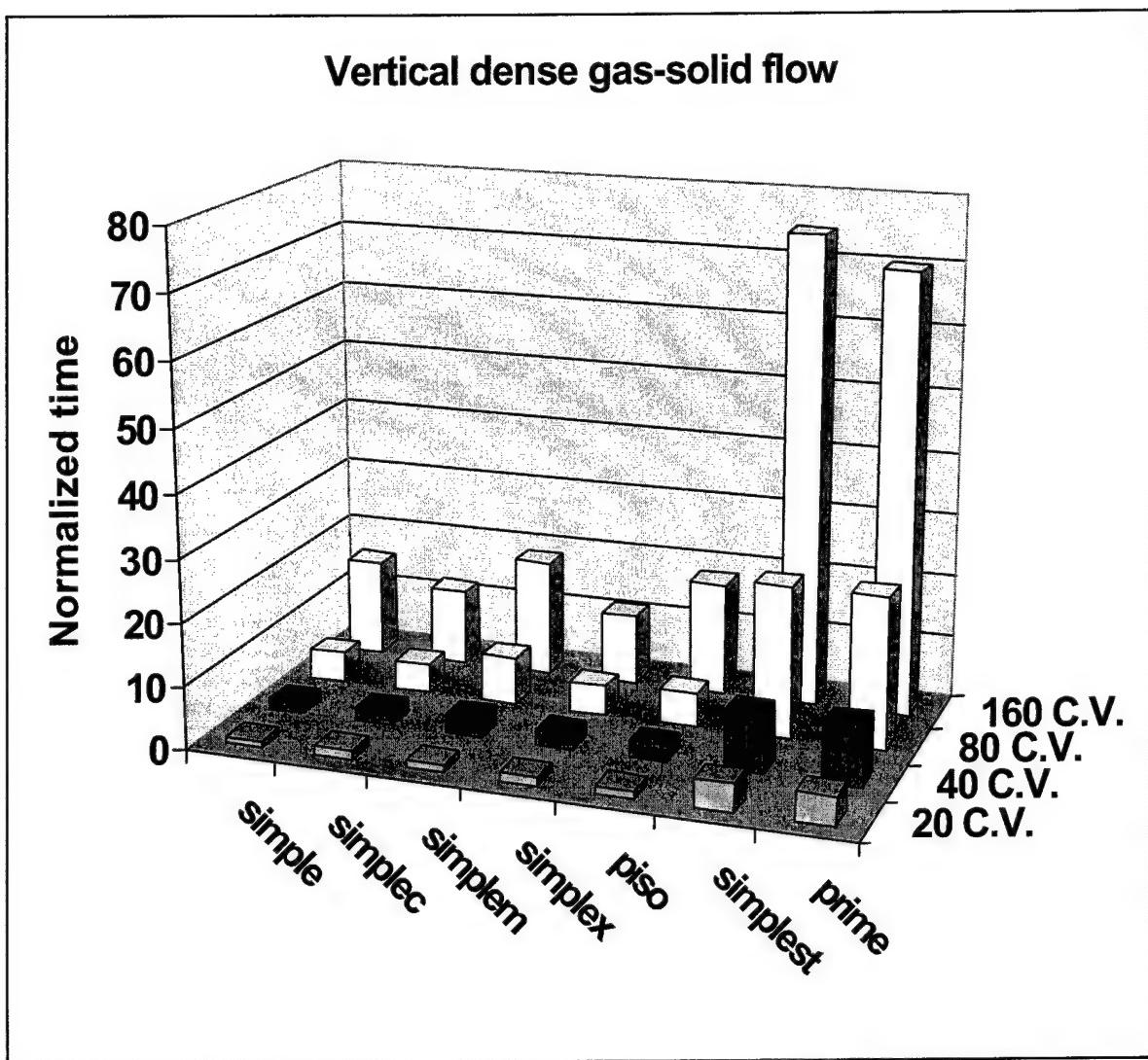


Fig. 14 Normalized CPU-times for the vertical dense gas-solid flow problem.

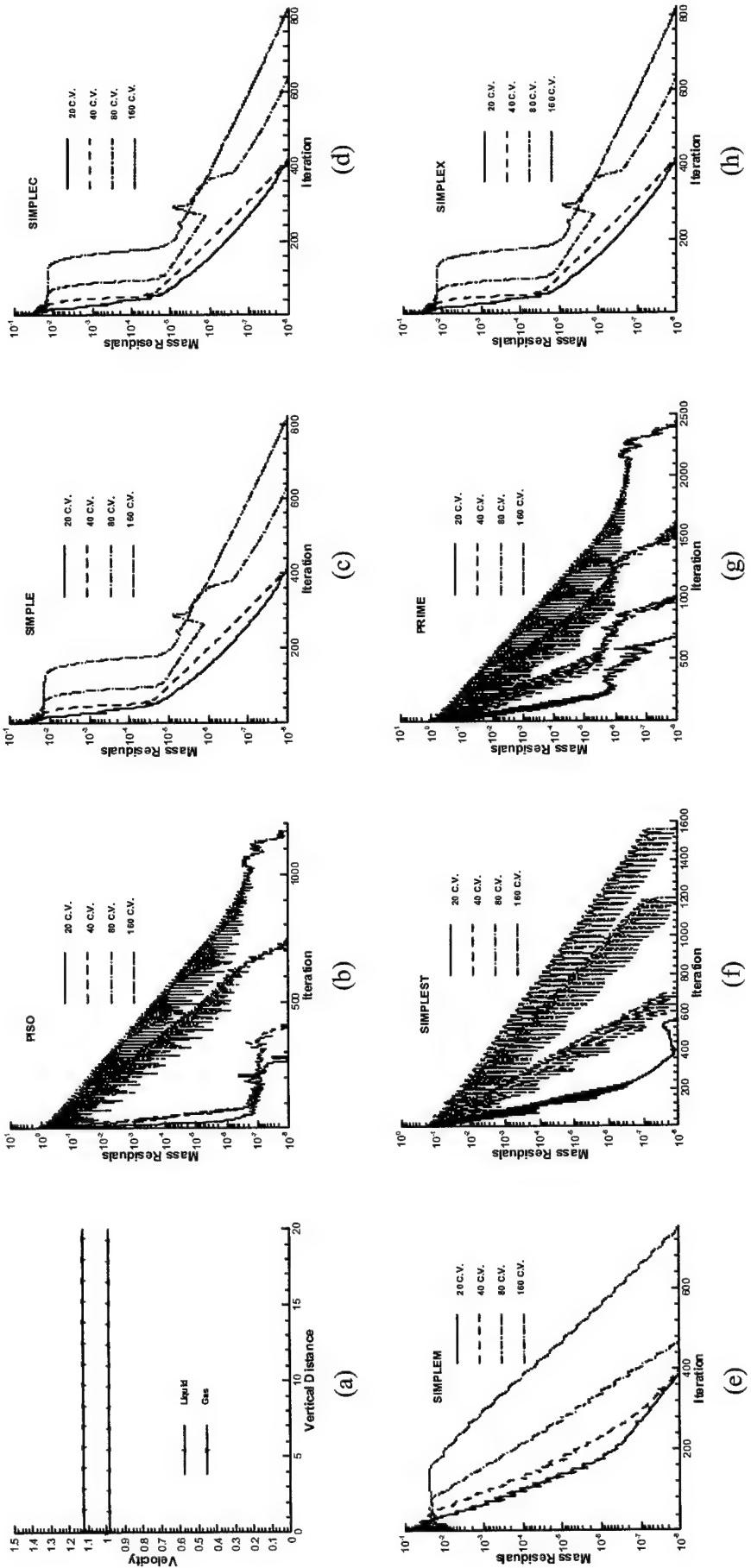


Fig. 15 (a) gas and particle velocity distributions; and (b)-(h) convergence histories on the different grid systems for the vertical dilute bubbly flow problem.

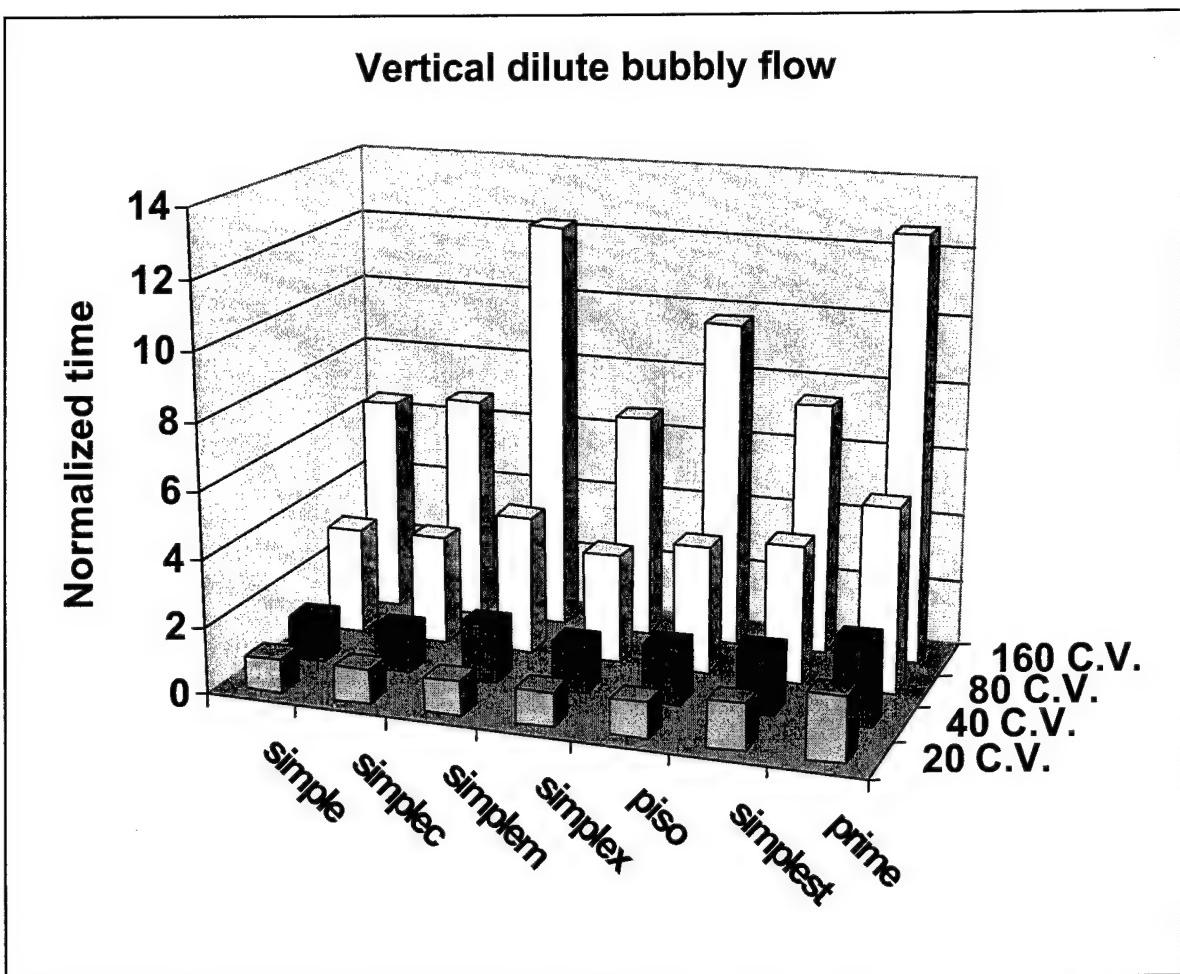
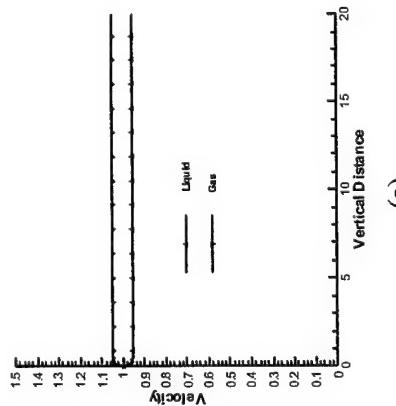
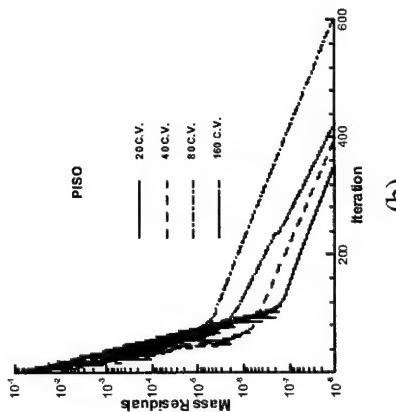


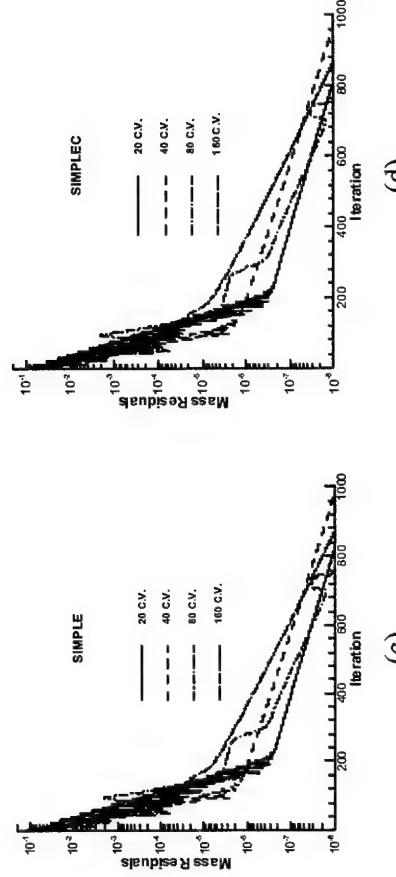
Fig. 16 Normalized CPU-times for the vertical dilute bubbly flow problem.



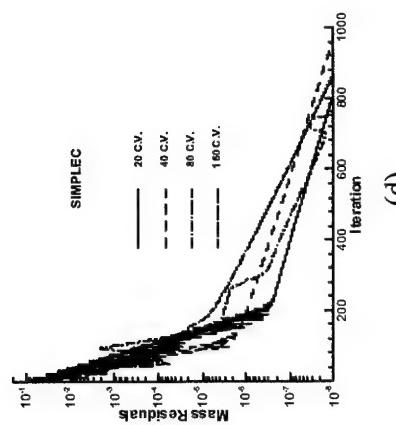
(a)



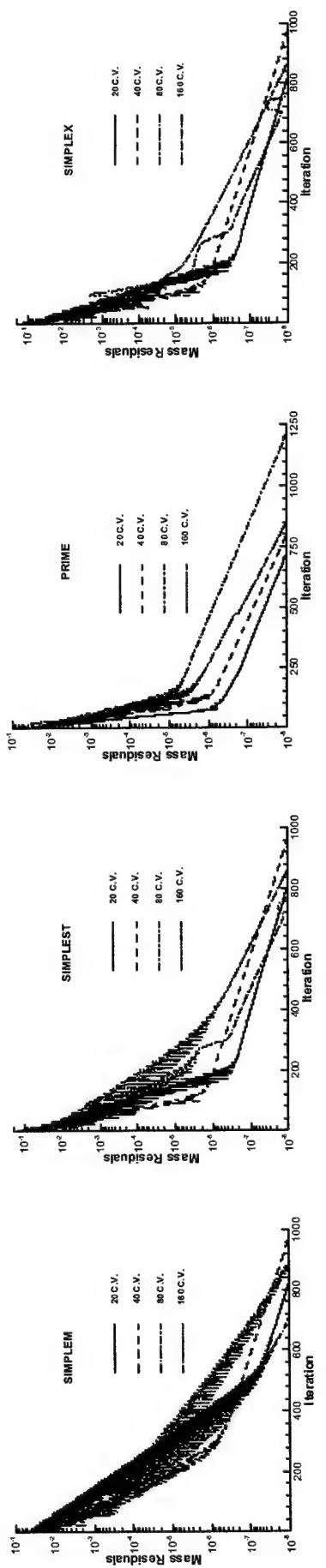
(b)



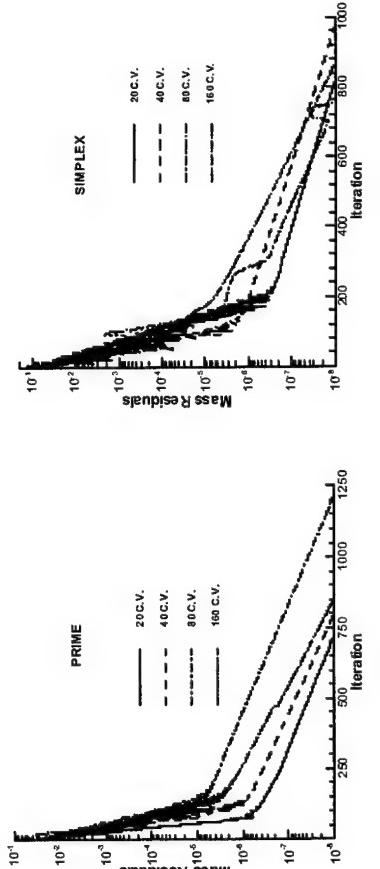
(c)



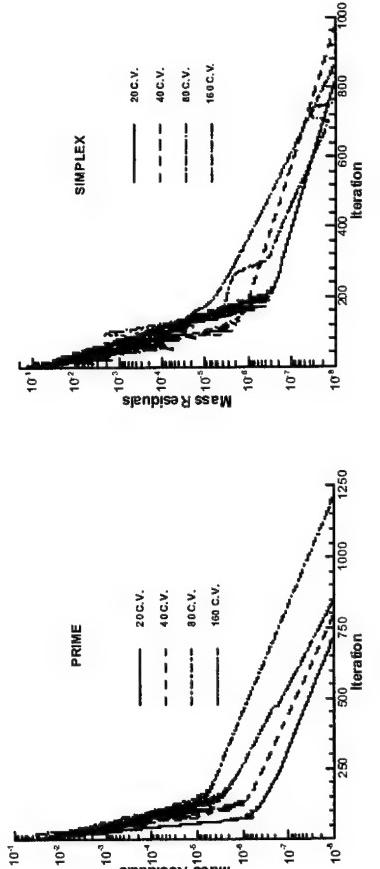
(d)



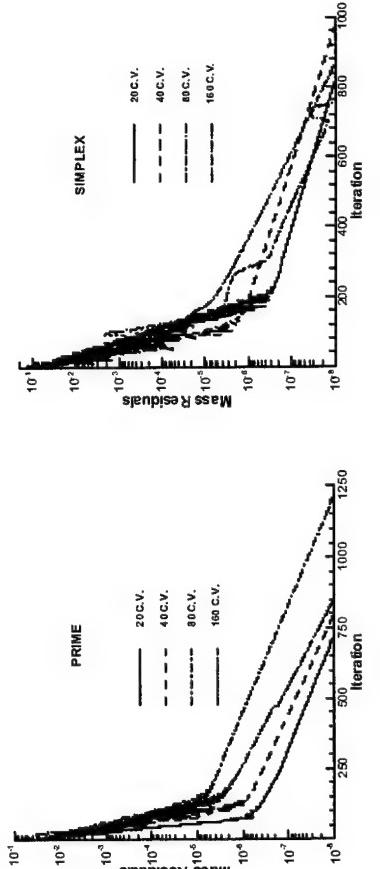
(e)



(f)



(g)



(h)

Fig. 17 (a) gas and particle velocity distributions, and (b)-(h) convergence histories on the different grid systems for the vertical dense bubbly flow problem.

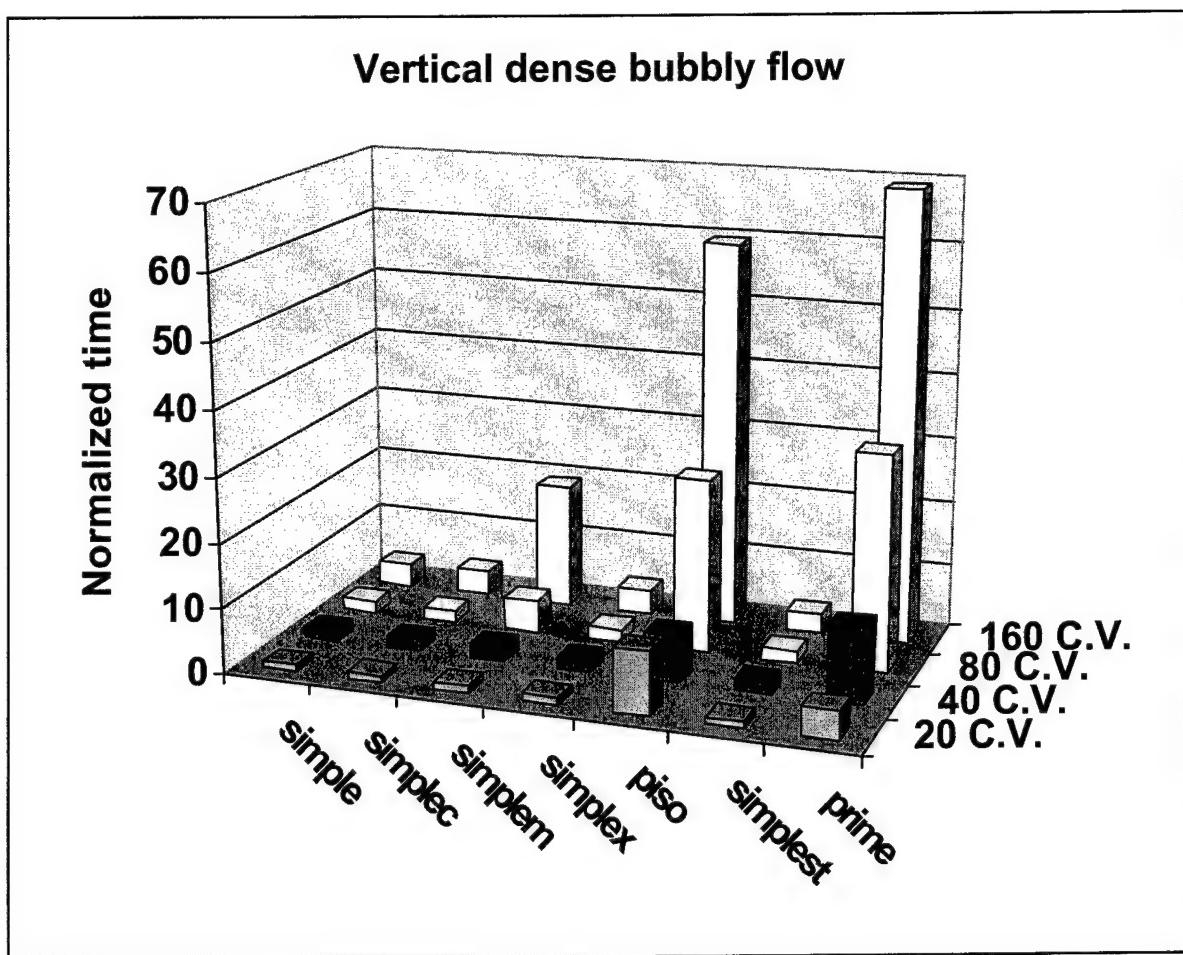
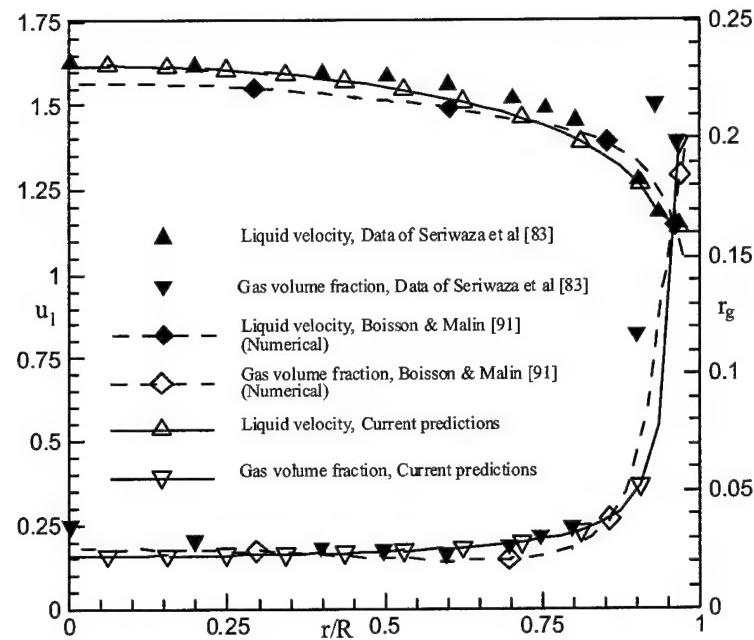
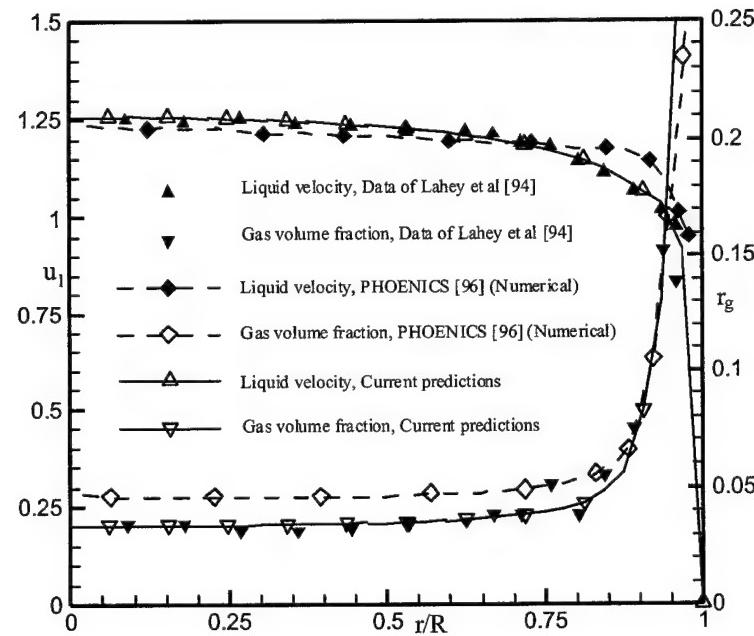


Fig. 18 Normalized CPU-times for the vertical dense bubbly flow problem.



(a)



(b)

Fig. 19 Comparison of fully developed liquid velocity and void fraction profiles for turbulent upward bubbly flow in a pipe; (a) Seriwaza et al. data, (b) Lahey et al. data.

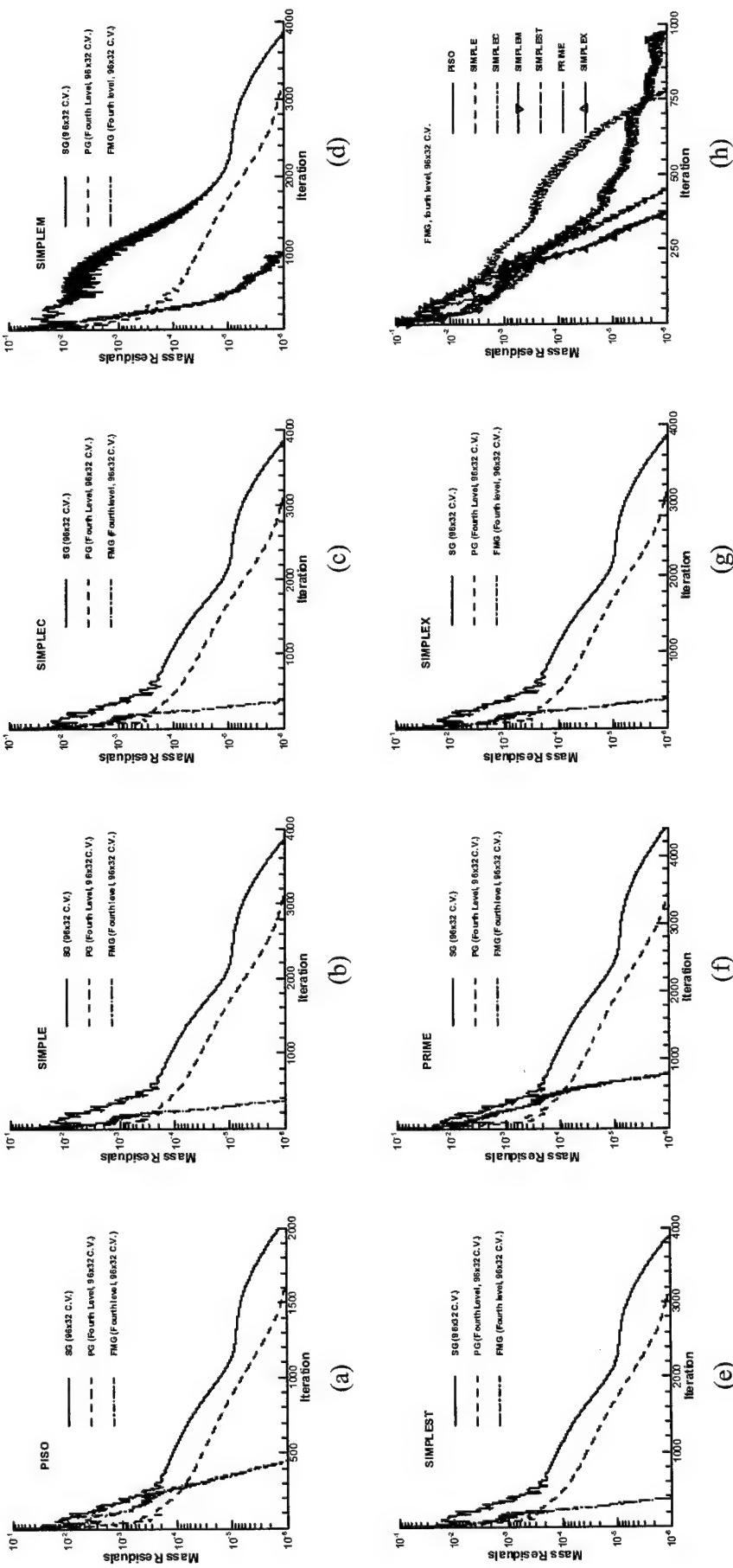


Fig. 20 (a)-(g) Convergence histories of the SG, PG, and FMG methods on the finest grid, and (h) convergence histories of the various algorithms on the finest mesh using the FMG method for turbulent upward bubbly flow in a pipe.

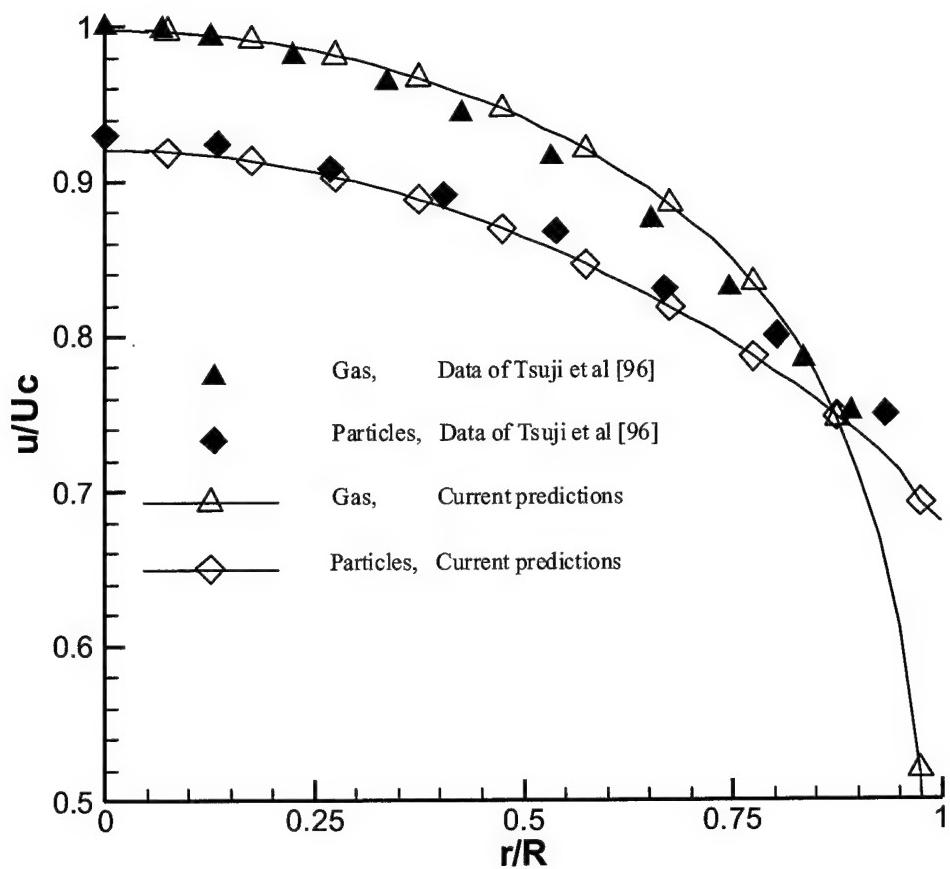


Fig. 21 Comparison of fully developed gas and particle velocity profiles for turbulent air-particle flow in a pipe.

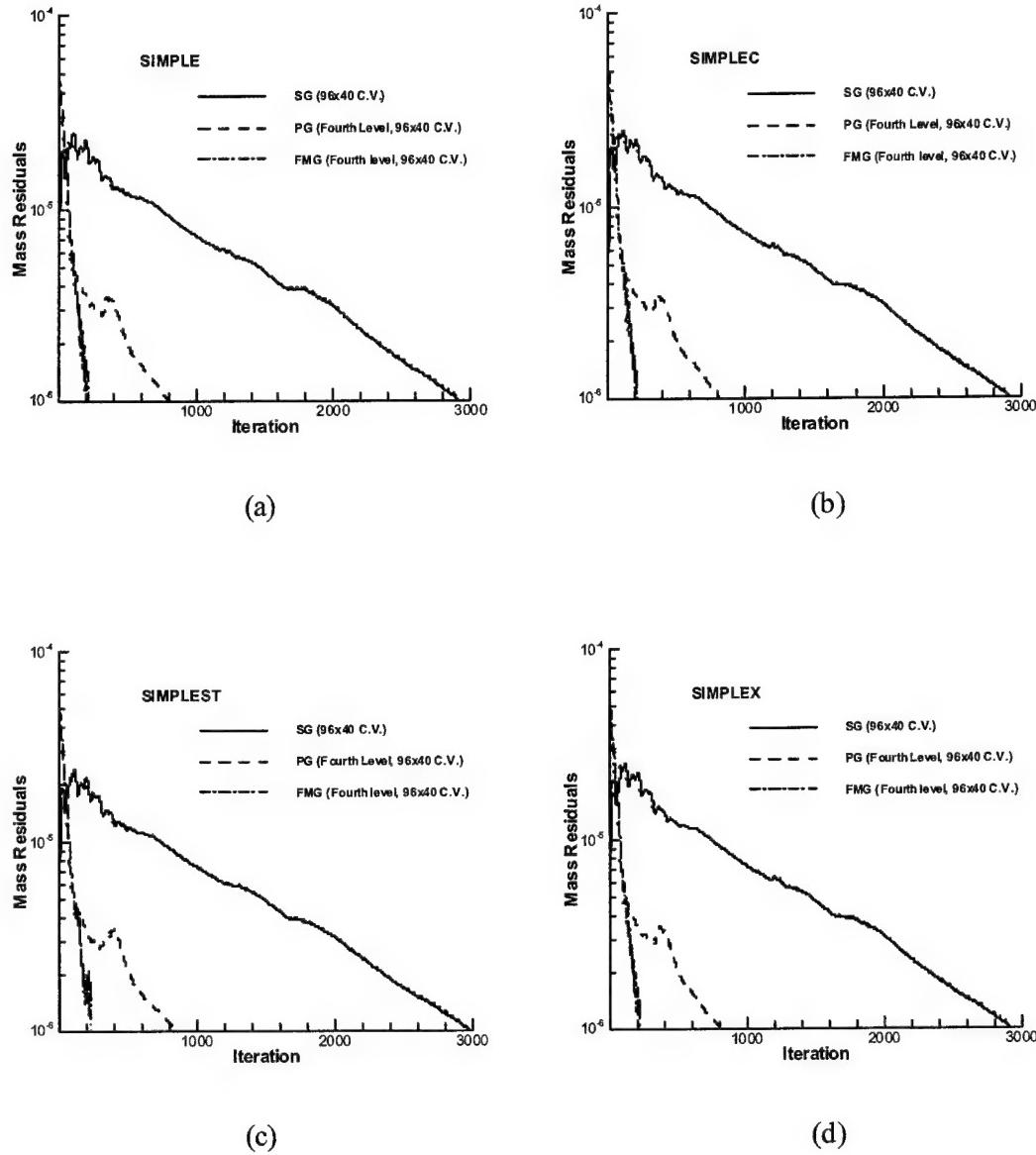


Fig. 22 Convergence histories of the (a) SIMPLE, (b) SIMPLEC, (c) SIMPLEST, and (d) SIMPLEX algorithms using the SG, PG, and FMG methods on the finest mesh for turbulent air-particle flow in a pipe.

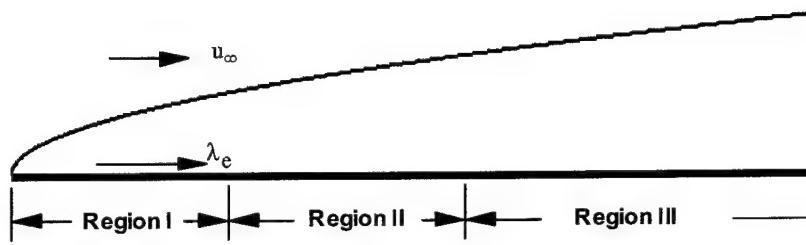


Fig. 23 The three different regions within the boundary layer of dusty flow over a flat plate.

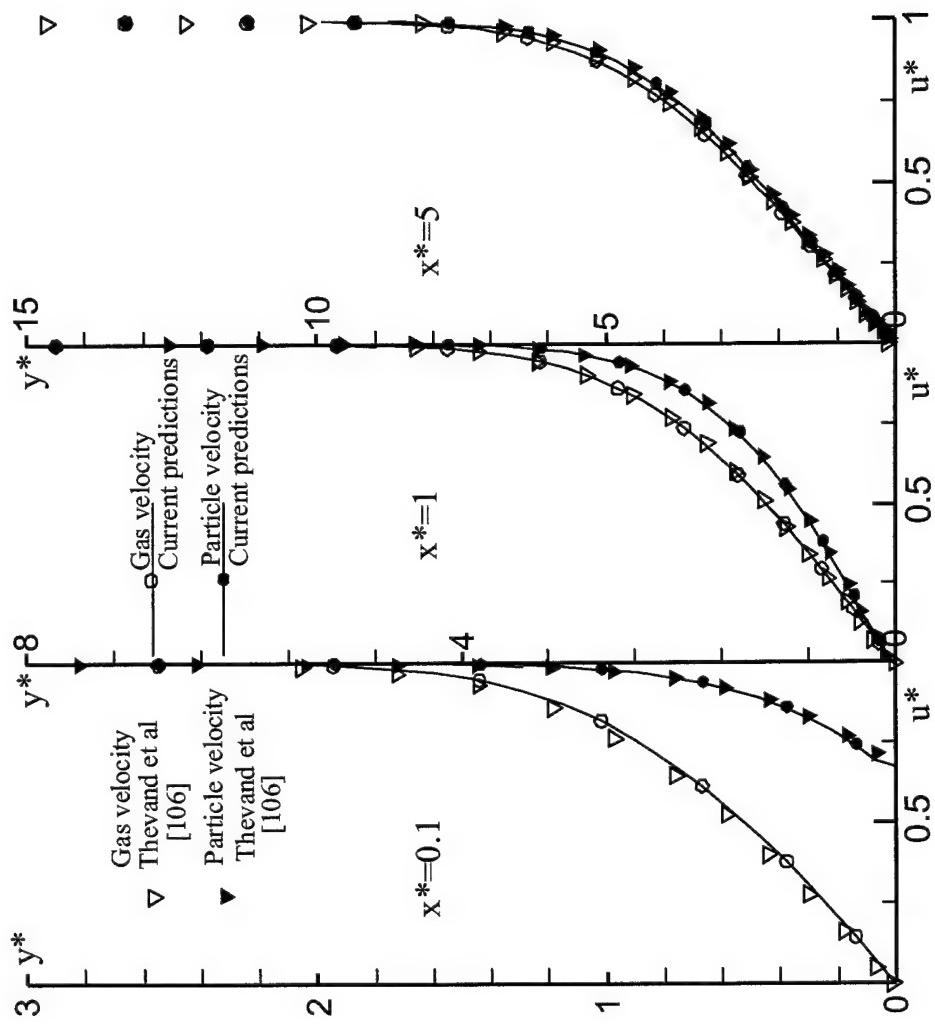


Fig. 24 Comparison of fully developed gas and particle velocity profiles inside the boundary layer at different axial locations for dilute two-phase flow over a flat plate

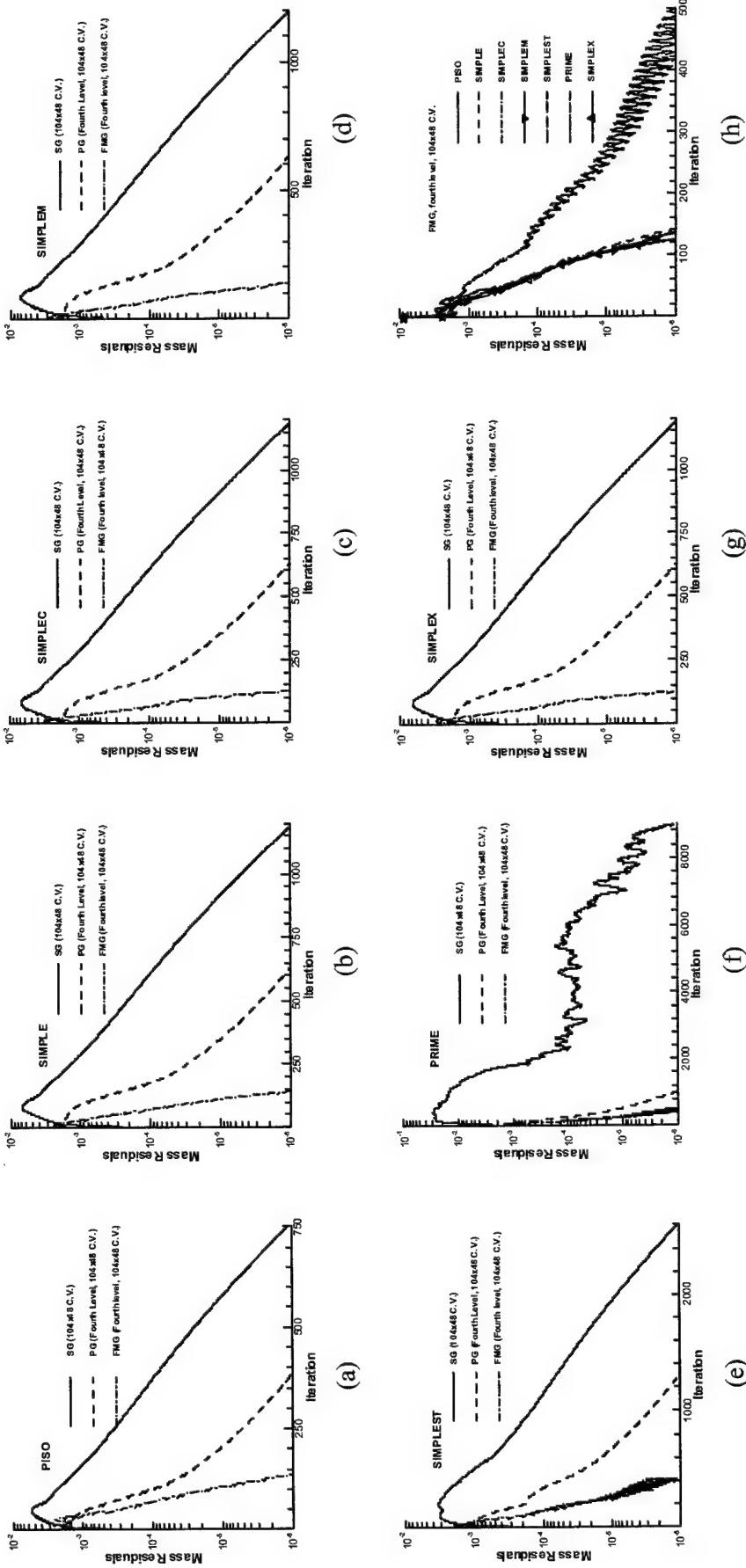


Fig. 25 (a)-(g) Convergence histories of the SG, PG, and FMG methods on the finest grid, and (h) convergence histories of the various algorithms on the finest mesh using the FMG method for dusty gas flow over a flat plate.

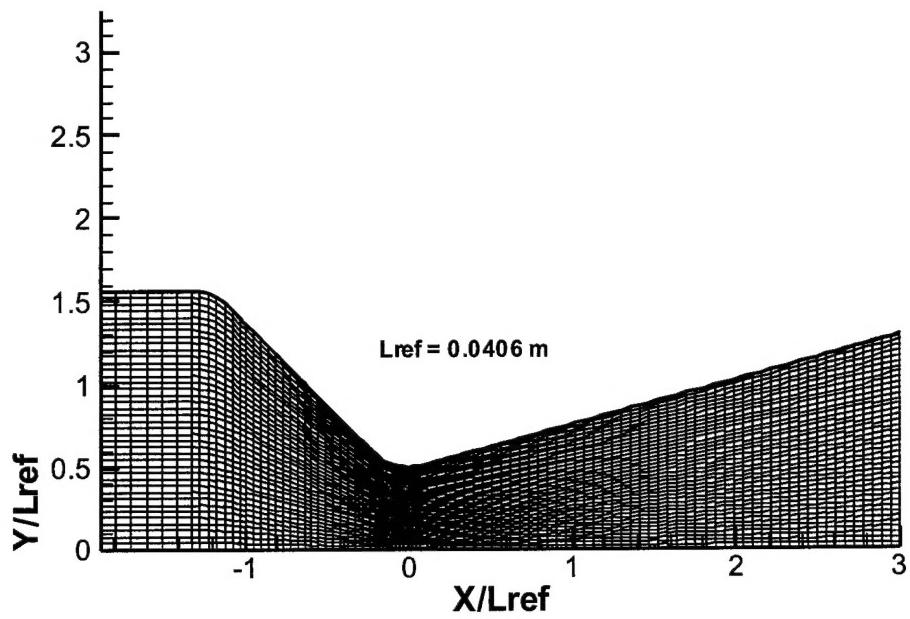
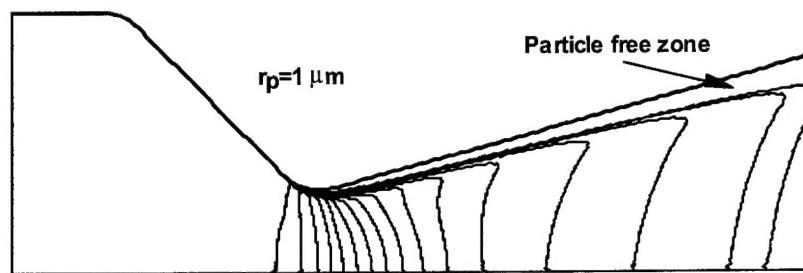
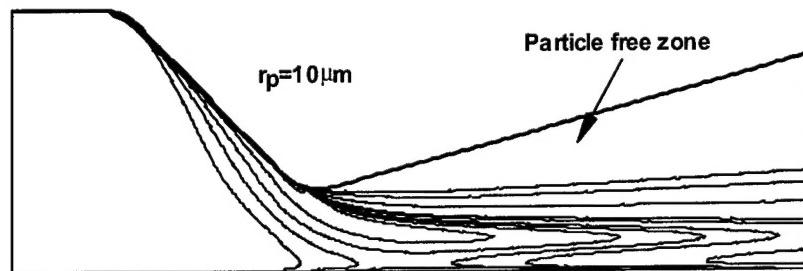


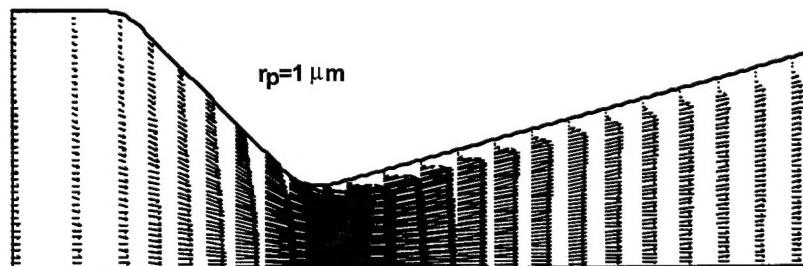
Fig. 26 Physical domain for the dusty gas flow in a converging-diverging nozzle.



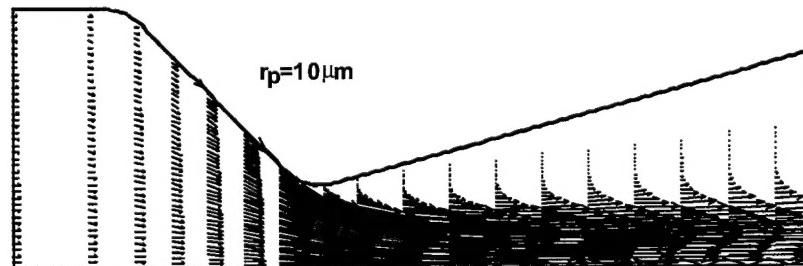
(a)



(b)

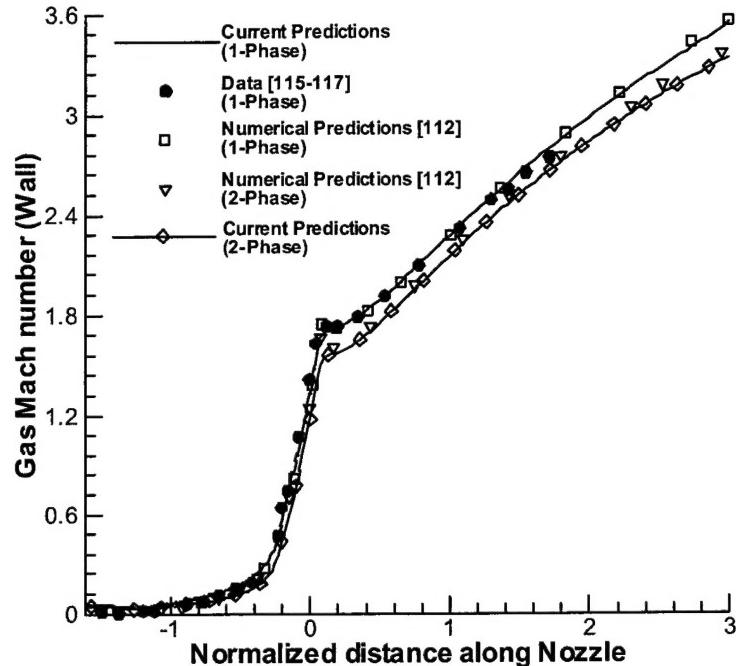


(c)

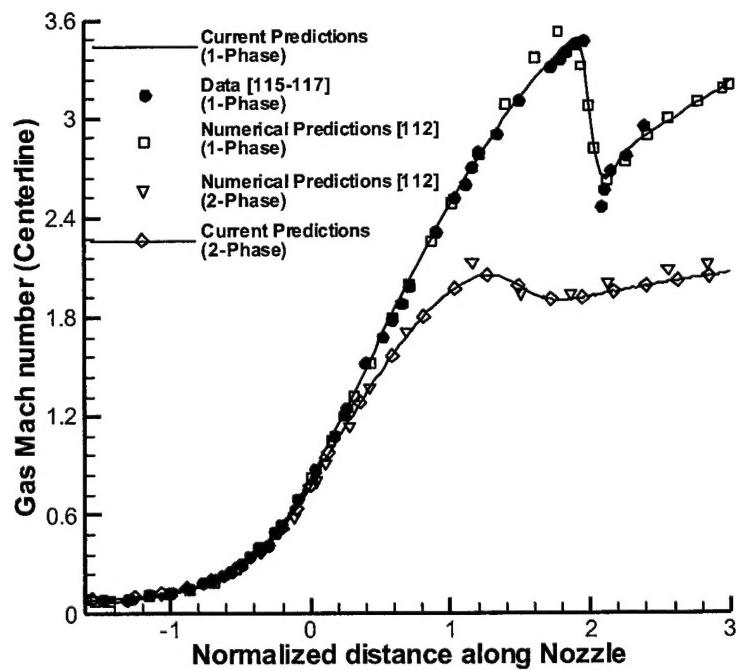


(d)

Fig. 27 (a,b) Volume Fraction contours and (c,d) particle velocity vectors for dusty gas flow in a converging-diverging nozzle.



(a)



(b)

Fig. 28 Comparison of one-phase and two-phase gas Mach number distributions along the (a) wall and (b) centerline of the dusty flow in a converging-diverging nozzle problem.

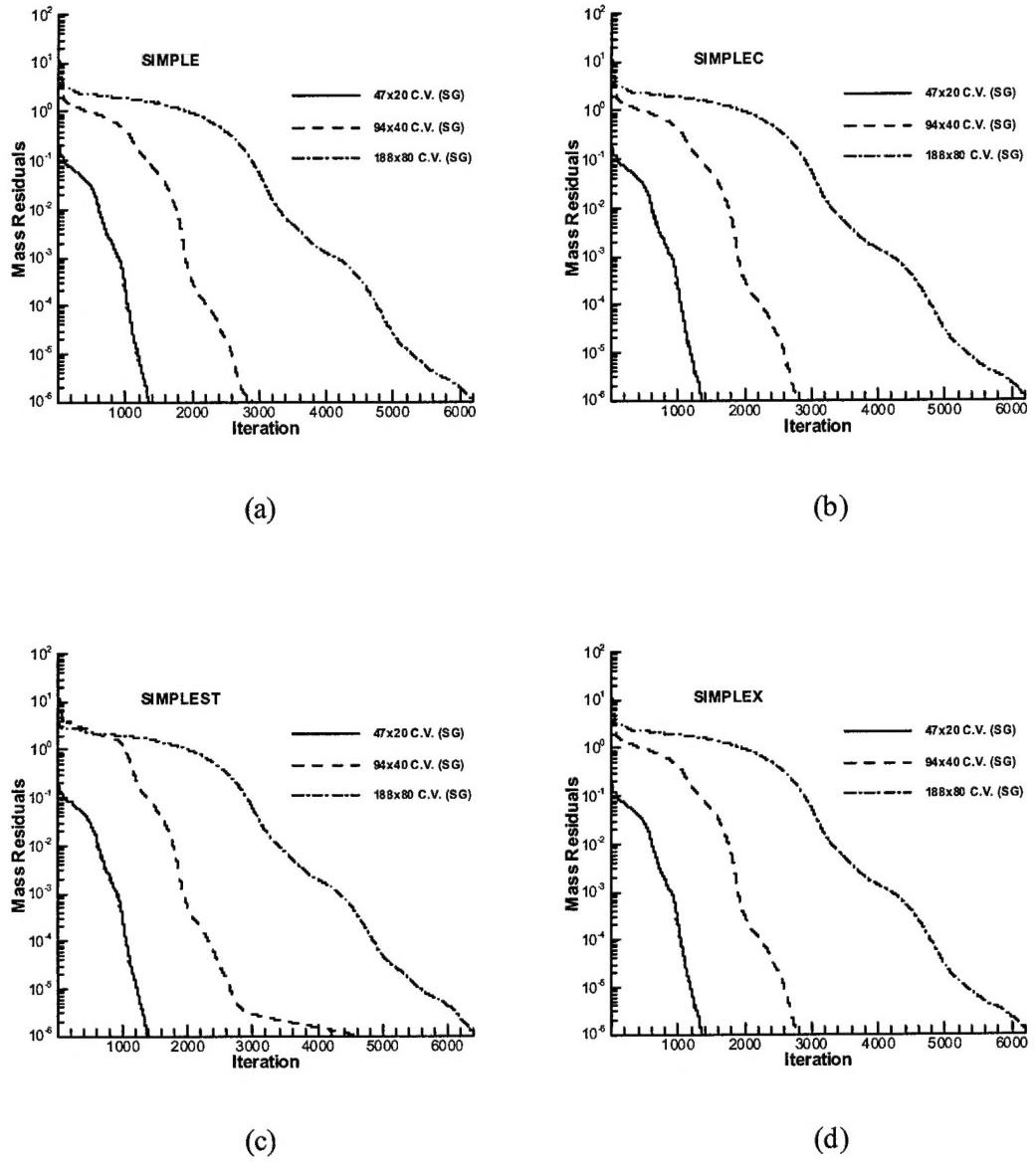


Fig. 29 Convergence histories of the (a) SIMPLE, (b) SIMPLEC, (c) SIMPLEST, and (d) SIMPLEX algorithms using the SG method for dusty gas flow in a converging-diverging nozzle.